

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО»**

**ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ**

**Кафедра системного програмування і спеціалізованих комп'ютерних систем**

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_  
(підпис) Тарасенко В.П.  
(ініціали, прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 201\_ р.

**Дипломний проект  
на здобуття ступеня бакалавра**

з напрямку підготовки **6.050102 «Комп'ютерна інженерія»**

на тему: Веб-орієнтований додаток моніторингу користувачів для адміністраторів

Виконав: студент IV курсу, групи KB-51

Катюха Богдан Миколайович

\_\_\_\_\_  
(підпис)

Керівник ст. викладач кафедри СПіСКС Наливайчук М.В.

\_\_\_\_\_  
(підпис)

Нормоконтроль к.т.н., доцент каф. СПіСКС Клятченко Я. М.

\_\_\_\_\_  
(підпис)

Рецензент ст. викладач кафедри ОТ Виноградов Ю.М.

\_\_\_\_\_  
(підпис)

Засвідчую, що у цьому дипломному проекті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2019 року

**Національний технічний університет України  
“Київський політехнічний інститут  
імені Ігоря Сікорського”**

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп’ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки 6.050102 «Комп’ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ В. П. Тарасенко

“ \_\_\_\_ ” \_\_\_\_\_ 2019 р.

**ЗАВДАННЯ**

**на дипломний проект студенту**

Катюсі Богдану Миколайовичу

1. Тема проекту «Веб-орієнтований додаток моніторингу користувачів для адміністраторів»,  
керівник проекту Наливайчук Микола Васильович, ст.викладач,  
затверджені наказом НМК „ІПО” КПІ імені Ігоря Сікорського від «06» квітня 2019 року № 1108-с.
2. Термін подання студентом проекту: «29» травня 2019 р.
3. Вихідні дані до проекту:
  - програмне забезпечення, яке дозволяє спростити роботу адміністраторів сайту.
4. Зміст пояснювальної записки:
  - аналіз існуючих рішень та обґрунтування теми дипломного проекту;
  - обґрунтування вибору інструментарію та засобів реалізації;
  - архітектура та опис реалізації дипломного проекту;
  - рекомендації для подальшого вдосконалення
5. Перелік графічного матеріалу:
  - поля статичних даних продаж (схема структурна);
  - API відображення замовлень (схема структурна);
  - маршрутизація по сайту (схема структурна);
  - API відображення веб-аналітики (схема структурна).

## 6. Консультанти:

| Питання       | Прізвище, ініціали та посада консультанта  | Підпис, дата   |                  |
|---------------|--|----------------|------------------|
|               |  | завдання видав | завдання прийняв |
| Нормоконтроль | Клятченко Я.М., к.т.н., доцент каф. СПіСКС |                |                  |

7. Дата видачі завдання: “31” жовтня 2018 р.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів виконання дипломної роботи         | Термін виконання етапів роботи | Примітка |
|-------|---|--------------------------------|----------|
| 1.    | Вивчення літератури за тематикою проекту        | 18.11.2018                     |          |
| 2.    | Розроблення та узгодження технічного завдання   | 01.12.2018                     |          |
| 3.    | Аналіз існуючих рішень                          | 16.12.2018                     |          |
| 4.    | Підготовка матеріалів першого розділу проекту   | 11.01.2019                     |          |
| 5.    | Розроблення програмного забезпечення            | 09.02.2019                     |          |
| 6.    | Відлагодження програмного продукту              | 24.02.2019                     |          |
| 7.    | Підготовка матеріалів другого розділу проекту   | 24.03.2019                     |          |
| 8.    | Розроблення інтерфейсу програмного забезпечення | 16.04.2019                     |          |
| 9.    | Підготовка графічної частини                    | 04.05.2019                     |          |
| 10.   | Оформлення документації дипломного проекту      | 20.05.2019                     |          |

Студент \_\_\_\_\_ Катюха Б.М.

Керівник проекту \_\_\_\_\_ Наливайчук М. В.

## АНОТАЦІЯ

Об'єкт розробки – веб-орієнтований додаток для відслідковування різних показників існуючого сайту, збору і візуалізації даних для адміністраторів сайту.

Додаток включає в себе: відображення різноманітних графіків та діаграм за різними фільтрами про користувачів сайту.

Програмне забезпечення допоможе швидко та якісно аналізувати користувачів сайту за багатьма показниками. Додаток дозволяє зменшити трудозатратність і збільшити швидкість вирішення деяких задач адміністраторів сайту. Окрім даних про користувачів програма відображає статистичні дані сайту що аналізується.

Програма була виконана на мові програмування JavaScript із застосуванням бібліотеки React.

Ключові слова:

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ВІДСЛІДКОВУВАННЯ ПОКАЗНИКІВ САЙТУ ТА ЇХ КОРИСТУВАЧІВ, JAVASCRIPT, REACT

## **ANNOTATION**

The object of development - a web-based application for tracking various indicators of an existing site, collection and visualization of data for site administrators.

The application includes: displaying various graphs and charts based on different users' filters for the site.

The software will help you quickly and qualitatively analyze users of the site by many indicators. The application allows you to reduce labor costs and increase the speed of solving some tasks of site administrators.

In addition to user data, the program displays the statistics of the site being analyzed. The program was implemented in the JavaScript programming language using the React library.

Keywords:

**SOFTWARE OF RESPONSIBILITIES OF THE SITE INDICATORS AND THEIR  
USERS, JAVASCRIPT, REACT INDICATORS**

| Поз. | Формат | Позначення         | Найменування  | Кількість | Прим. |
|------|--------|--------------------|---|-----------|-------|
|      |        |                    | <u>Документація загальна</u>  |           |       |
|      |        |                    | <u>Новорозроблена</u>   |           |       |
|      | A4     | ІАЛЦ.045440.002 ТЗ | Веб-орієнтований додаток контролю користувачів для адміністраторів. | 3         |       |
|      |        |                    | Технічне завдання.  |           |       |
|      |        |                    |   |           |       |
|      | A4     | ІАЛЦ.045440.003 ТП | Веб-орієнтований додаток контролю користувачів для адміністраторів. | 1         |       |
|      |        |                    | Відомість технічного проекту.                                       |           |       |
|      |        |                    |   |           |       |
|      | A4     | ІАЛЦ.045440.004 ПЗ | Веб-орієнтований додаток контролю користувачів для адміністраторів. | 53        |       |
|      |        |                    | Пояснювальна записка.   |           |       |
|      |        |                    |   |           |       |
|      | A4     | ІАЛЦ.045440.005 Д1 | Веб-орієнтований додаток контролю користувачів для адміністраторів. | 1         |       |
|      |        |                    | Поля статистичних даних продаж.                                     |           |       |
|      |        |                    | Схема структурна.   |           |       |
|      |        |                    |   |           |       |
|      | A4     | ІАЛЦ.045440.006 Д2 | Веб-орієнтований додаток контролю користувачів для адміністраторів. | 1         |       |
|      |        |                    | API відображення замовлень.   |           |       |
|      |        |                    | Схема структурна.   |           |       |
|      |        |                    |   |           |       |
|      | A4     | ІАЛЦ.045440.007 Д3 | Веб-орієнтований додаток контролю користувачів для адміністраторів. | 1         |       |
|      |        |                    | Маршрутизація по сайту.   |           |       |
|      |        |                    | Схема структурна.   |           |       |
|      |        |                    |   |           |       |
|      |        |                    |   |           |       |
|      |        |                    |   |           |       |

|           |      |                 |       |      |  |  |       |         |  |
|-----------|------|-----------------|-------|------|--|--|-------|---------|--|
|           |      |                 |       |      | <b>ІАЛЦ.045440.001 ОА</b>  |  |       |         |  |
| Зм.       | Арк. | № докум.        | Підп. | Дата | Веб-орієнтований додаток моніторингу користувачів для адміністраторів<br><b>Опис альбому</b> | Літ.                                     | Аркуш | Аркушів |  |
| Розроб.   |      | Катюха Б.М.     |       |      |  |  |       |         |  |
| Перевір.  |      | Наливайчук М.В. |       |      |  |  | 1     | 2       |  |
|           |      |                 |       |      |  | КПІ ім. Ігоря Сікорського,<br>ФПМ, КВ-51 |       |         |  |
| Н. контр. |      | Клятченко Я.М.  |       |      |  |  |       |         |  |
| Затв.     |      | Тарасенко В.П.  |       |      |  |  |       |         |  |

[illegible]

|    |      |          |       |      |                           |      |
|----|------|----------|-------|------|---------------------------|------|
|    |      |          |       |      | <b>ІАЛЦ.045440.001 ОА</b> | Арк. |
| 3м | Лист | № докум. | Підп. | Дата |                           | 2    |

## ЗМІСТ

|   |   |
|---|---|
| 1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ.....                                 | 2 |
| 2. ПІДСТАВА ДЛЯ РОЗРОБКИ .....  | 2 |
| 3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ .....                                      | 2 |
| 4. ДЖЕРЕЛА РОЗРОБКИ.....  | 2 |
| 5. ТЕХНІЧНІ ВИМОГИ.....   | 2 |
| 5.1. Вимоги до програмного продукту, що розробляється .....             | 2 |
| 5.2. Вимоги до апаратного забезпечення.....                             | 3 |
| 5.3. Вимоги до програмного та апаратного забезпечення користувача ..... | 3 |
| 6. ЕТАПИ РОЗРОБКИ .....   | 4 |

|                  |             |                 |              |             |  |  |  |             |  |               |  |
|------------------|-------------|-----------------|--------------|-------------|--|--|--|-------------|--|---------------|--|
|                  |             |                 |              |             | <b>ІАЛЦ. 045440.002 ТЗ</b>   |  |  |             |  |               |  |
|                  |             |                 |              |             |  |  |  |             |  |               |  |
| <b>Зм</b>        | <b>Лист</b> | <b>№ докум.</b> | <b>Підп.</b> | <b>Дата</b> |  |  |  |             |  |               |  |
| <b>Розроб.</b>   | Катюха      |                 |              |             | <div>Веб-орієнтований додаток моніторингу користувачів для адміністраторів</div> <div><b>Технічне завдання</b></div> |  |  |             |  |               |  |
| <b>Перев.</b>    | Наливайчук  |                 |              |             |  |  |  |             |  |               |  |
|                  |             |                 |              |             |  |  |  |             |  |               |  |
| <b>Н. контр.</b> | Клятченко   |                 |              |             |  |  |  |             |  |               |  |
| <b>Затв.</b>     | Тарасенко   |                 |              |             |  |  |  |             |  |               |  |
|                  |             |                 |              |             | <b>Літ.</b>  |  |  | <b>Лист</b> |  | <b>Листів</b> |  |
|                  |             |                 |              |             |  |  |  | 1           |  | 4             |  |
|                  |             |                 |              |             | <b>КПІ ім. Ігоря Сікорського,<br/>ФПМ, КВ-51</b>   |  |  |             |  |               |  |



## 1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Веб-орієнтований додаток моніторингу користувачів для адміністраторів».

Галузь застосування: організація прийому та обробки вхідних даних на обслуговування IT-інфраструктури.

## 2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання дипломного проекту ступеня «бакалавр комп'ютерної інженерії», затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

## 3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є створення веб-орієнтованого додатку для контролю та відображення інформації про користувачів існуючого сайту для полегшення роботи адміністраторів сайту.

## 4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

## 5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- сумісність з будь якою операційною системою (IOS,Android,Linux);

|           |             |                 |              |             |                           |                  |
|-----------|-------------|-----------------|--------------|-------------|---------------------------|------------------|
|           |             |                 |              |             | <b>ІАЛЦ.045440.002 ТЗ</b> | <b>Лист</b><br>2 |
| <b>Зм</b> | <b>Лист</b> | <b>№ докум.</b> | <b>Підп.</b> | <b>Дата</b> |                           |                  |

- можливість фільтрування даних;
- створення зручного інтерфейсу;
- відображення сторінки аналітики;
- можливість відображення графіків та діаграм;

## 5.2. Вимоги до апаратного забезпечення

- Процесор: 2,4-ядерний, MediaTek, Snapdragon, Kirin, Intel Atom;
- Оперативна пам'ять: 2 Гб;
- Наявність доступу до мережі Internet;

## 5.3. Вимоги до програмного та апаратного забезпечення користувача

- Операційна система Windows, Linux, iOS;

|    |      |          |       |      |                           |      |
|----|------|----------|-------|------|---------------------------|------|
|    |      |          |       |      | <b>ІАЛЦ.045440.002 ТЗ</b> | Лист |
|    |      |          |       |      |                           | 2    |
| Зм | Лист | № докum. | Підп. | Дата |                           |      |

## 6. ЕТАПИ РОЗРОБКИ

| № з/п | Назва етапів виконання дипломного проекту                | Термін виконання етапів |
|-------|--|-------------------------|
| 1.    | Вивчення літератури за тематикою проекту                 | 15.10.2018              |
| 2.    | Розроблення та узгодження технічного завдання            | 30.11.2018              |
| 3.    | Аналіз існуючих рішень                                   | 18.12.2018              |
| 4.    | Підготовка матеріалів першого розділу дипломного проекту | 10.02.2019              |
| 5.    | Підготовка матеріалів другого розділу дипломного проекту | 18.03.2019              |
| 6.    | Підготовка графічної частини дипломного проекту          | 05.04.2019              |
| 7.    | Оформлення документації дипломного проекту               | 27.05.2019              |
| 8.    | Попередній огляд матеріалів диплому на кафедрі           | 29.05.2019              |

| Поз. | Формат | Позначення         | Найменування   | Кількість | Прим. |
|------|--------|--------------------|--|-----------|-------|
|      |        |                    | <u>Документація загальна</u>   |           |       |
|      |        |                    | <u>Новорозроблена</u>  |           |       |
|      | A4     | ІАЛЦ.045440.004 ПЗ | Веб-орієнтований додаток контролю користувачів для адміністраторів . | 53        |       |
|      |        |                    | Пояснювальна записка   |           |       |
|      |        |                    |  |           |       |
|      | A4     | ІАЛЦ.045440.005 Д1 | Веб-орієнтований додаток контролю користувачів для адміністраторів.  | 1         |       |
|      |        |                    | Поля статистичних даних продаж .                                     |           |       |
|      |        |                    | Схема структурна.  |           |       |
|      |        |                    |  |           |       |
|      | A4     | ІАЛЦ.045440.006 Д2 | Веб-орієнтований додаток контролю користувачів для адміністраторів . | 1         |       |
|      |        |                    | API відображення замовлень .   |           |       |
|      |        |                    | Схема структурна.  |           |       |
|      |        |                    |  |           |       |
|      | A4     | ІАЛЦ.045440.007 Д3 | Веб-орієнтований додаток контролю користувачів для адміністраторів . | 1         |       |
|      |        |                    | Маршрутизація по сайту.  |           |       |
|      |        |                    | Схема структурна .   |           |       |
|      |        |                    |  |           |       |
|      | A4     | ІАЛЦ.045440.008 Д4 | Веб-орієнтований додаток контролю користувачів для адміністраторів . | 1         |       |
|      |        |                    | API відображення веб-аналітики.                                      |           |       |
|      |        |                    | Схема структурна .   |           |       |
|      |        |                    |  |           |       |
|      |        | Диск CD-ROM        | Матеріали дипломного проекту   | 1         |       |
|      |        |                    |  |           |       |
|      |        |                    |  |           |       |
|      |        |                    |  |           |       |
|      |        |                    |  |           |       |

|           |                |          |       |      |  |  |  |  |  |  |  |  |  |  |
|-----------|----------------|----------|-------|------|--|--|--|--|--|--|--|--|--|--|
|           |                |          |       |      | <b>ІАЛЦ.045440.003 ТП</b>  |  |  |  |  |  |  |  |  |  |
| Зм.       | Арк.           | № докум. | Підп. | Дата | <div>Веб-орієнтований додаток моніторингу користувачів для адміністраторів</div> <div>Відомість технічного проекту</div>                       |  |  |  |  |  |  |  |  |  |
| Розроб.   | Катюха Б.М.    |          |       |      |  |  |  |  |  |  |  |  |  |  |
| Перевір.  | Наливайчук М.В |          |       |      |  |  |  |  |  |  |  |  |  |  |
|           |                |          |       |      |  |  |  |  |  |  |  |  |  |  |
| Н. контр. | Клятченко Я.М. |          |       |      |  |  |  |  |  |  |  |  |  |  |
| Затв.     | Тарасенко В.П. |          |       |      | <div>Літ.</div> <div>Аркуш</div> <div>Аркушів</div> <div></div> <div>1</div> <div>1</div> <div>КПІ ім. Ігоря Сікорського,<br/>ФПМ, КВ-51</div> |  |  |  |  |  |  |  |  |  |

## ЗМІСТ

|   |    |
|---|----|
| ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ                            | 3  |
| ВСТУП   | 6  |
| Розділ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ | 8  |
| 1.1. Загальний опис проблем адміністратора сайту                          | 8  |
| 1.2. Аналіз існуючих рішень для даної задачі                              | 11 |
| 1.2.1. Програма для відображення та адміністрування даних ngx-admin:      | 12 |
| 1.2.2. Програма для адміністрування Adminator:                            | 13 |
| 1.2.3. Програма для розміщення та відображення інформації Joomla:         | 14 |
| 1.2.4. Система для керування наповненням сайту WordPress:                 | 15 |
| Розділ 2. ОБГРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТАРІЮ ТА ЗАСОБІВ РЕАЛІЗАЦІЇ       | 16 |
| 2.1. Вибір операційної системи для розробки                               | 16 |
| 2.2. Обґрунтування вибору мови програмування                              | 17 |
| 2.3. Сторонні засоби для покращення розробки                              | 21 |
| 2.3.1. Вибір основної бібліотеки React                                    | 21 |
| 2.3.2. Технологія JSX   | 25 |
| 2.3.3. Вибір інструменту для написання стилів                             | 26 |
| 2.4. Середовище розробки Visual Studio Code                               | 27 |

|                  |             |                 |              |             |  |  |  |  |  |             |               |   |    |
|------------------|-------------|-----------------|--------------|-------------|--|--|--|--|--|-------------|---------------|---|----|
|                  |             |                 |              |             | <b>ІАЛЦ.045440.004 ПЗ</b>  |  |  |  |  |             |               |   |    |
|                  |             |                 |              |             |  |  |  |  |  |             |               |   |    |
| <b>Зм</b>        | <b>Лист</b> | <b>№ докум.</b> | <b>Підп.</b> | <b>Дата</b> |  |  |  |  |  |             |               |   |    |
| <b>Розроб.</b>   |             | Катюха          |              |             | <i>Веб-орієнтований додаток моніторингу користувачів для адміністраторів</i> |  |  |  | <b>Літ.</b>                                      | <b>Лист</b> | <b>Листів</b> |   |    |
| <b>Перев.</b>    |             | Наливайчук      |              |             |  |  |  |  |  |             |               | 1 | 53 |
|                  |             |                 |              |             |  |  |  |  | <b>КПІ ім. Ігоря Сікорського,<br/>ФПМ, КВ-51</b> |             |               |   |    |
| <b>Н. контр.</b> |             | Клятченко       |              |             |  |  |  |  |  |             |               |   |    |
| <b>Затв.</b>     |             | Тарасенко       |              |             | <b>Пояснювальна записка</b>  |  |  |  |  |             |               |   |    |

Розділ 3.    АРХІТЕКТУРА ТА ОПИС РЕАЛІЗАЦІЇ РОЗРОБЛЕНОГО  
ПЗ 30

Розділ 4.    РЕКОМЕНДАЦІЇ ДЛЯ ПОДАЛЬШОГО  
ВДОСКОНАЛЕННЯ \_\_\_\_\_ 45  
ВИСНОВОКИ \_\_\_\_\_ 49  
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ \_\_\_\_\_ 52

### **ДОДАТКИ**

#### **Додаток 1. Копії графічних матеріалів:**

- ІАЛЦ.045440.005 Д1. Поля статичних даних продаж. Схема структурна;
- ІАЛЦ.045440 0.06 Д2. АРІ відображення замовлень. Схема структурна;
- ІАЛЦ.045440.007 Д3. Маршрутизація по сайту . Схема структурна;
- ІАЛЦ.04544 0.08 Д4. АРІ відображення веб-аналітики. Схема структурна.

#### **Додаток 2. Лістинг програмного коду**

#### **Додаток 3. Презентація**

|           |             |                 |              |             |                           |             |
|-----------|-------------|-----------------|--------------|-------------|---------------------------|-------------|
|           |             |                 |              |             | <b>ІАЛЦ.045440.004 ПЗ</b> | <b>Лист</b> |
| <b>Зм</b> | <b>Лист</b> | <b>№ докум.</b> | <b>Підп.</b> | <b>Дата</b> |                           | <b>2</b>    |

## ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ОС – операційна система

ПЗ – програмне забезпечення

API – Application Programming Interface - прикладний програмний інтерфейс

LTS – Long Term Support

JS – Java Script – веб-орієнтована скриптова мова програмування

ES – EcmaScript - стандарт мови програмування, затверджений міжнародною організацією ECMA згідно зі специфікацією ECMA-262. Найвідомішими реалізаціями стандарту є мови JavaScript, JScript та ActionScript, які широко використовується у Вебі[1]

AJAX - Asynchronous JavaScript And XML — підхід до побудови користувацьких інтерфейсів веб-застосунків, за яких веб-сторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і сама звідти довантажує потрібні користувачу дані. AJAX — один з компонентів концепції DHTML[2]

JSON (укр. запис об'єктів JavaScript, англ. JavaScript Object Notation) — це текстовий формат обміну даними між комп'ютерами. JSON базується на тексті, може бути прочитаним людиною. Формат дозволяє описувати об'єкти та інші структури даних. Цей формат головним чином використовується для передачі структурованої інформації через мережу (завдяки процесу, що називають серіалізацією)[3]

MVC - Модель–вигляд–контролер (або Модель–представлення–контролер, англ. Model-view-controller) — архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення[4]

CSS (англ. Cascading Style Sheets, укр. Каскадні таблиці стилів) — спеціальна мова, що використовується для опису зовнішнього вигляду сторінок, написаних мовами розмітки даних[5]

Sass - (англ. Syntactically Awesome Stylesheets) — скриптова метамова, яка інтерпретується в каскадні таблиці стилів (CSS)[6] .

LESS - Leaner Style Sheets, компактна таблиця стилів

JSX - JavaScript XML – розширення синтаксиса мови програмування JavaScript

VS Code – Visual Studio Code – редактор коду

SPA – single-page application – односторінковий додаток

Scss - Sassy CSS – вид синтаксиса Sass

HTML - Hypertext Markup Language, укр. Мова розмітки гіпертекстових документів —стандартна мова розмітки для створення веб-сторінок і веб-додатків . З Cascading Style Sheets (CSS) і JavaScript, вона утворює триаду основних технологій для World Wide Web.[7]

HTTP — Hyper Text Transfer Protocol, протокол передачі гіпер-текстових документів - протокол передачі даних, що використовується в комп'ютерних мережах.[8]

PHP - Hypertext Preprocessor (укр. гіпертекстовий препроцесор)— скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера.[9]

Git — розподілена система керування версіями файлів та спільної роботи.[10]

Фреймворк (від англ. Framework, каркас, платформа) – інфраструктура програмних рішень, що полегшує розробку складних систем. Спрощено дану інфраструктуру можна вважати своєрідною комплексною бібліотекою.[11]

DOM - Об'єктна модель документа (англ. Document Object Model) — специфікація прикладного програмного інтерфейсу для роботи зі



структурованими документами (як правило, документами XML). Визначається ця специфікація консорціумом W3C.[12]

W3C - Консорціум Всесвітньої павутини (англ. World Wide Web Consortium) — головна міжнародна організація, що розробляє й впроваджує технологічні стандарти для всесвітньої павутини.[13]

GNU – вільна, UNIX-подібна ОС.

XML - розширювана мова розмітки (англ. Extensible Markup Language) — запропонований консорціумом World Wide Web Consortium стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема, через Інтернет.[14]

ASP.NET — технологія створення веб-застосунків і веб-сервісів від компанії Майкрософт.[15]

|    |      |          |       |      |                           |      |
|----|------|----------|-------|------|---------------------------|------|
|    |      |          |       |      | <b>ІАЛЦ.045440.004 ПЗ</b> | Лист |
| Зм | Лист | № докум. | Підп. | Дата |                           | 5    |

## ВСТУП

Основною метою даної дипломної роботи є розробка веб-додатка контролю користувачів сайту для адміністраторів сайту. Їх часто називають менеджерами або керівниками проекту. Важливість їхньої роботи важко переоцінити, адже вони виконують широкий спектр задач. Для полегшення та збільшення ефективності їхньої роботи було розроблено програмне забезпечення.

Розробка не є чимось новим, але існуючі аналоги мають як плюси так і мінуси, в проекті було використано їх переваги, а недоліки було виправлено. На виході отримали функціональний, зручний та простий інструмент для адміністраторів сайту.

Для комфортного вирішення проблем було розроблено веб-орієнтовану програму, яка дозволить реалізовувати перегляд аналіз та пошук потрібної інформації.

Веб-додаток використовує графіки, таблиці та діаграми для кращого сприйняття даних.

Програма забезпечує економію часу та ресурсів, звільнюючи від виконання аналогічних задач вручну. Це покращить роботоспроможність основного продукту, адже ПЗ безпосередньо впливає на якість обслуговування користувачів. Додаток може бути використаним для зворотнього зв'язку з клієнтами.

Застосування програмного забезпечення може використовуватися для будь-якого сайту, де необхідно відображувати та аналізувати інформацію. Для цього необхідно реалізувати сервіси доступу до бази даних, для отримання та запису відповідної інформації.

Сьогодні ж, починаючи з маленьких компаній, кожен хоче опанувати мережу Інтернет для розширення клієнтської бази, та збільшення прибутків. Виникає необхідність створення веб-сайтів, які потрібно обслуговувати та

|    |      |          |       |      |                           |      |
|----|------|----------|-------|------|---------------------------|------|
|    |      |          |       |      | <b>ІАЛЦ.045440.004 ПЗ</b> | Лист |
| Зм | Лист | № докум. | Підп. | Дата |                           | 6    |

моніторити. Тема дипломного проекту включає в себе такі функції. Тому веб-додаток має бути доступним для середнього та, навіть, малого бізнесу, адже їм не потрібно буде витрачати ресурси на власні реалізації аналогічних продуктів.

Виконання було здійснено сучасними засобами розробки, які забезпечують конкурентоздатність продукту.

Також було сформовано можливі майбутні способи удосконалення програмного забезпечення.

|                  |                    |                        |                     |                    |                                  |                    |
|------------------|--------------------|------------------------|---------------------|--------------------|----------------------------------|--------------------|
|                  |                    |                        |                     |                    | <b><i>ІАЛЦ.045440.004 ПЗ</i></b> | <b><i>Лист</i></b> |
| <b><i>Зм</i></b> | <b><i>Лист</i></b> | <b><i>№ докум.</i></b> | <b><i>Підп.</i></b> | <b><i>Дата</i></b> |                                  | <b><i>7</i></b>    |

# Розділ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

## 1.1. Загальний опис проблем адміністратора сайту

Адміністратора сайту часто використовують та включають до відділу менеджменту та керівництва. Це говорить про те, що він виконує різноманітні функції в які входять:

- аналіз результатів роботи ресурсу;
- перевірка коректної роботи;
- впровадження нового функціоналу;
- збільшення рейтингу сайту;

Ці функції дуже складні для виконання і водночас важливі для роботи сайту. Часто необхідно запрошувати сторонніх осіб для допомоги адміністратору.[16]

В перелік робочих обов'язків та навичок адміністратора сайту входить дуже багато різноманітних задач, оскільки він має вміти займатися тестуванням, працювати з даними, та займатися розвитком проекту. Також, в залежності від компанії адміністратор може виконувати функцію підтримки користувачів та аналізу даних. В тому числі він може адмініструвати системи збору та відображення різноманітних даних, на основі яких робити певні висновки та практичні дії. Виходячи з цього, адміністратор має вирішувати широкий перелік задач.

Зазвичай адміністратору важко впоратися з такою великою кількістю проблем, тому для забезпечення більш комфортної роботи адміністратора і було обрано відповідну тему дипломної роботи. Це дозволить йому самостійно, не залучаючи сторонніх осіб, виконувати широкий спектр обов'язків.

|    |      |          |       |      |                           |      |
|----|------|----------|-------|------|---------------------------|------|
|    |      |          |       |      | <b>ІАЛЦ.045440.004 ПЗ</b> | Лист |
| Зм | Лист | № докум. | Підп. | Дата |                           | 8    |

Проблеми які вирішується за допомогою розробленого програмного забезпечення входять до переляку проблем адміністраторів сайтів. В його обов'язки входить зберігання та відображення даних для подальшої роботи з ними.

Одна з проблем – несанкціонований доступ до даних. Ця проблема досить важлива, адже інформація про користувачів, різні статистичні дані не повинні розголошуватися. І будь-який доступ до заборонених даних може спричинити серйозні проблеми.

Поширення конфіденційних даних може викликати шквал обурення користувачів, та, навіть, юридичні проблеми. З цим потрібно бути дуже обережним, і надавати доступ для адміністрування тільки окремим особам.

Робота з великими об'ємами даних також важливий компонент роботи адміністратора. Оскільки в мережі Інтернет кількість користувачів дуже велика і стрімко зростає, кількість даних збільшується також. Для опрацювання всієї інформації без автоматизації потрібно витратити багато часу, що також є проблемою. Але, навіть, при автоматизації збору даних, їх потрібно правильно і повноцінно використовувати.

Кількість даних може бути настільки великою, що для того аби в них розібратися і дістати потрібну інформацію також затрачається багато зусиль. Щоб уникнути цієї проблеми слід грамотно відображати дані. Наприклад, користувач телефонує в службу підтримки і запитує питання, відповідь на яке потребує деяких даних. В цей момент необхідно ідентифікувати користувача і знайти необхідну інформацію.

Ніхто не любить довго чекати, тому якщо це швидко виконувати не вдіється, в користувача можуть скластися негативні враження. Це може призвести до втрати як існуючих так і потенційних клієнтів.

Для відображення даних краще використовувати різні формати такі як: таблиці, графіки, діаграми. За допомогою такого відтворення можна вести статистику та на її базі робити певний аналіз.

|    |      |          |       |      |                           |      |
|----|------|----------|-------|------|---------------------------|------|
|    |      |          |       |      | <b>ІАЛЦ.045440.004 ПЗ</b> | Лист |
| Зм | Лист | № докум. | Підп. | Дата |                           | 9    |

Останнім часом різні статистичні дані користуються великим попитом. Оскільки статистика допомагає вести стратегію розвитку та розробки веб-сайту, це досить важливий компонент.

В умовах великої конкуренції необхідно проводити аналіз ринків для покращення роботи сайту та для збільшення користувачів. Аналіз аудиторії сайту може також стати одним з ключів успіху проекту, в залежності від віку, національної приналежності та інших показників, можна підлаштовувати веб-сайт.

Аналітична спроможність залежить від якості та повноти даних і від того наскільки зручно ними користуватись. Для цього необхідно реалізувати швидкий пошук та фільтрацію даних за різними параметрами. Це дозволить швидко та якісно аналізувати та редагувати дані і відслідковувати за кожним користувачем.

Для точної аналітики необхідно виключити людський фактор з деяких процесів. Це необхідно зробити для уникнення помилок які людина може спричинити через свою неуважність. Найчастіше неточності виникають при обчисленнях, оскільки окрім банальних помилок дані округлюються, це призводить до різниці реальних та статистичних показників. В цьому випадку може статися ефект метелика, коли одна незначна помилка може спричинити фатальні наслідки.

При недостовірній статистиці виникають проблеми з аналітикою, а це може стати на шляху до успіху. Якщо, аналітика рухається в неправильному напрямку, то розвиток сайту може стати на тупиковий шлях, і в жорстких умовах сучасної конкуренції проект може закритися.

Веб-додаток допоможе вирішувати різноманітні задачі з короткий період. Для кожної компанії час – це гроші, а система забезпечить велику їх економію.

Ще десять років тому сайти мали тільки великі компанії, адже Інтернет, на той час ще не набув такої популярності. Сьогодні ж, підприємницька діяльність починає своє функціонування з написання простих веб-сторінок. Оскільки більшість людей щоденно використовують Інтернет для купівлі, продажу, простого перегляду новин та інших цілей, то однією із заporук успіху проекту може стати грамотно реалізований сайт. Без системи адміністрування зробити це буде складно.

## **1.2. Аналіз існуючих рішень для даної задачі**

Існує багато рішень даної задачі, але проблема в тому що більшість з них вузько спеціалізовані під конкретні дані і мають досить складну навігацію пошуку потрібних даних. Також зустрічаються додатки з поганим інтерфейсом, що спричиняє труднощі в сприйнятті інформації. Часто на вирішення цих проблем не витрачається багато засобів та часу, аргументуючи це тим, що не ніяк не впливає на основний продукт. Але це є помилкою, оскільки для стрімкого розвитку та підтримки великих проектів необхідно мати якісну систему збору та відображення даних. Слід зазначити, що більшість потужних сайтів мають досить непогані засоби вирішення цих проблем, з деякими з них мені, навіть, доводилося працювати, але навести їх в якості прикладів неможна через комерційну таємницю.

Можна привести приклади програм які слугують шаблоном для вирішення даної задачі. Їх часто використовують компанії, не дивлячись на деякі недоліки. Ці шаблони досить одноманітні та часто виконані в схожому вигляді, але часто не пристосовані до вирішення практичних задач. Також вони значно відрізняються від реальних засобів для адміністрування, оскільки робиться акцент не на наповнення даними а на картинку, що може спричинити гарне враження, але згодом певне розчарування в практичному функціонування додатку.

### 1.2.1. Програма для відображення та адміністрування даних ngx-admin:

Виконано у вигляді односторінкового веб-сайту засобами TypeScript і Angular. Це сучасні та потужні інструменти написання додатків, але основною проблемою є складність інтегрування власного контенту в цю розгалужену систему.

Програмне забезпечення має багато інструментів для адміністрування (Рис. 1.). Здійснюється відображення статистичних даних в різному вигляді, що дозволяє якісно їх аналізувати. Можна встановлювати різні фільтри для відображення потрібної інформації, яка виводиться на екран у зручному вигляді для користувача. Існують засоби налаштування під конкретного користувача, для подальшого зручного користування.

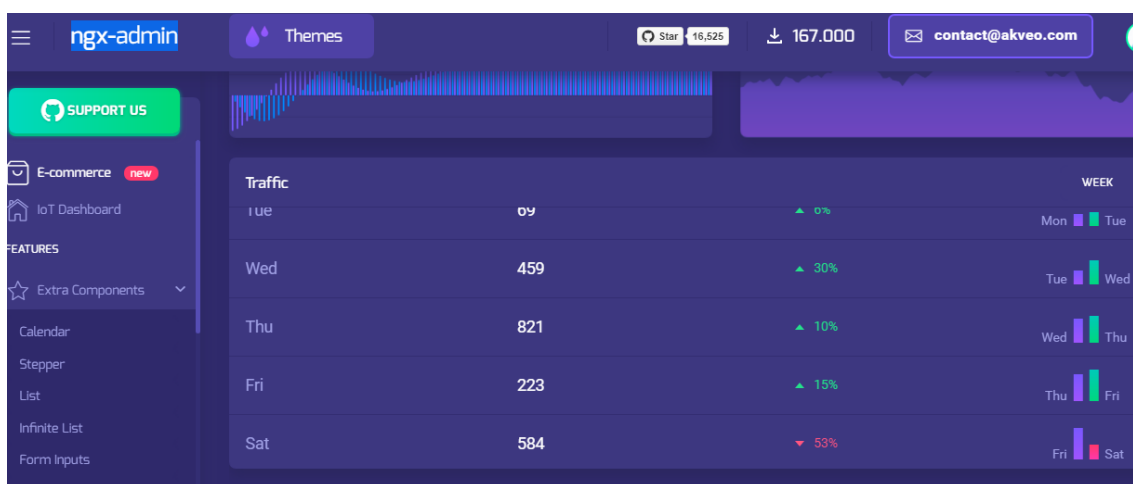


Рис. 1. Приклад реалізації ngx-admin

#### Переваги ПЗ:

- Велика кількість даних для аналізу;
- Швидкий та якісний пошук інформації
- Гарний дизайн відображення даних

#### Недоліки ПЗ:

- Невисока швидкодія;
- Дані відображаються але немає їхнього аналізу ;



- Це програмне забезпечення відображає надлишкову інформацію, яка є ускладненням для нового користувача.

### 1.2.2. Програма для адміністрування Adminator:

Виконано вигляді веб-сайту із застосуванням сучасного дизайну. Часто можна зустріти анімацію, графіки та, навіть, карти. Це справляє позитивне враження і досить приємно користуватися такою програмою. Але наскільки ефективна ця система?

Додаток виконано у вигляді шаблону з великим функціоналом (Рис. 2.), для подальшого налаштування під конкретні цілі. Проблема лишається в тому, що великий функціонал та гарне стильове оформлення не завжди забезпечують гарне практичне застосування. Також невідомо як буде себе проявляти така система при великих об'ємах даних.

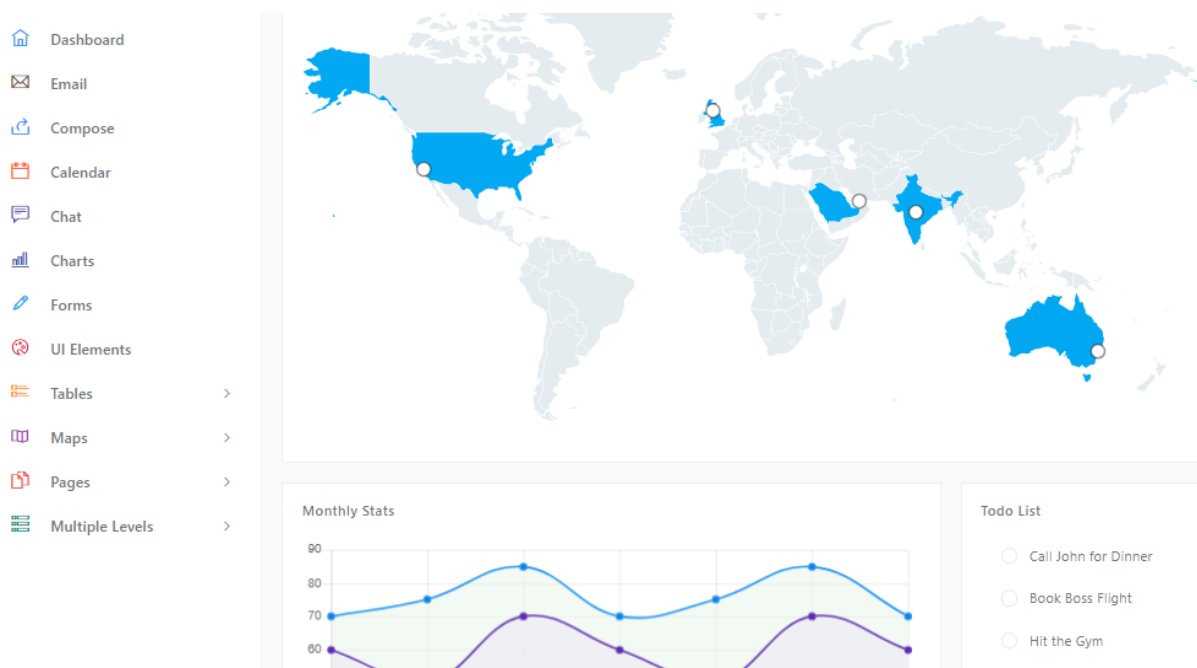


Рис. 2. Зовнішня реалізація програми Adminator

#### Переваги ПЗ:

- Великий функціонал;
- Сучасний дизайн;

### Недоліки ПЗ:

- Незручна панель фільтрування даних;
- Немає аналізу даних;
- Використовується невелика кількість інформації

### 1.2.3. Програма для розміщення та відображення інформації Joomla:

Joomla – засіб для управління наповненням та розмітки сайтів. Це відкритий продукт який написаний на мовах програмування JavaScript та PHP українськими розробниками.

В якості прикладу наведено систему, яка називається менеджер користувачів (Рис. 3.).

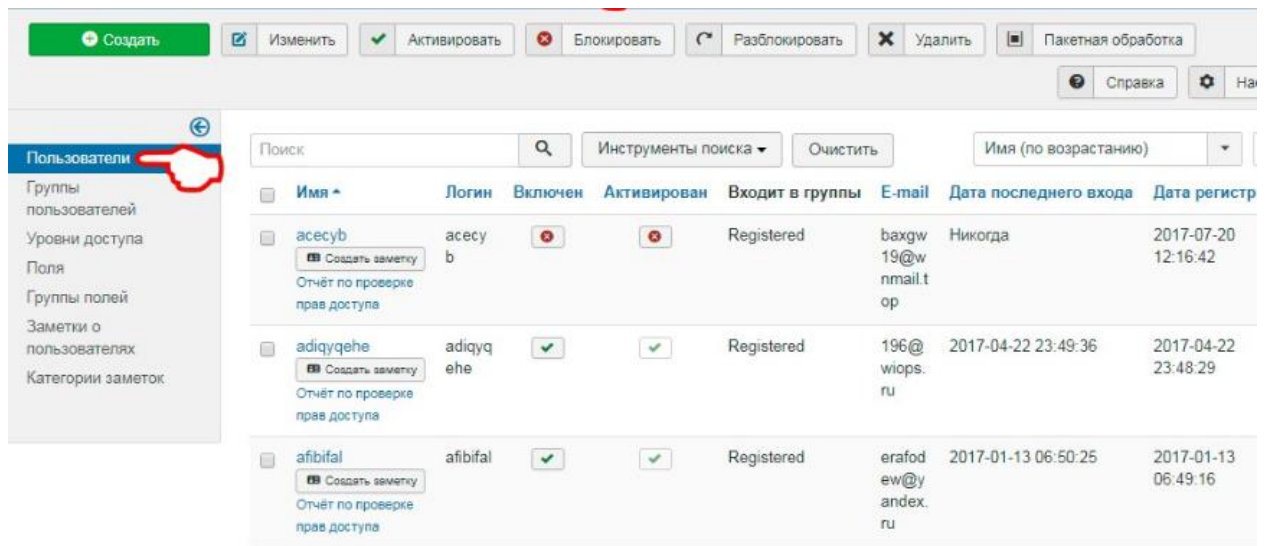


Рис. 3. Шаблон адміністрування даних реалізований за допомогою Joomla

### Переваги ПЗ:

- Простота використання;

### Недоліки ПЗ:

- Незручна для підлаштування під більш вузькоспеціалізовані задачі;
- Можуть виникати проблеми з використанням реальних даних;

#### 1.2.4. Система для керування наповненням сайту WordPress:

WordPress досить розповсюджена система, яка користується великою популярністю. Свою славу він здобув завдяки простоті використання та широкого спектру можливостей.

За допомогою різних шаблонів цієї системи можна також вирішувати задачі з якими стикаються адміністратори сайту, один з них показано на Рис.4.. Дозволяється використовувати різні теми, які можна підлаштовувати під себе для більшої зручності.

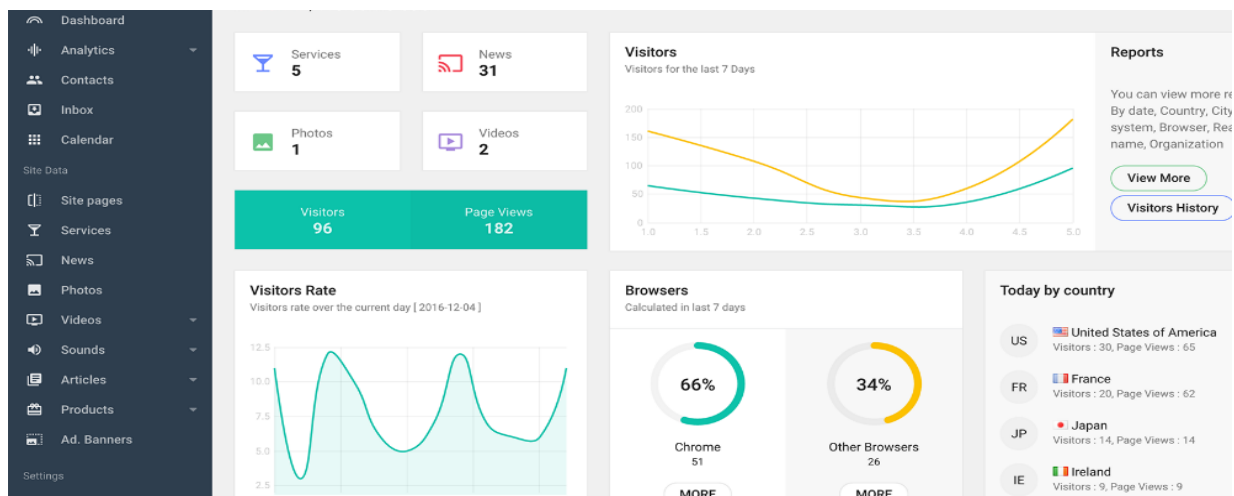


Рис. 4. Шаблон адміністрування даних реалізований за допомогою WordPress

#### Переваги ПЗ:

- Розповсюдженість на ринку;
- Різноманітність стилів

#### Недоліки ПЗ:

- По суті, являється тільки графічною оболонкою для відображення даних

## Розділ 2. ОБҐРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТАРІЮ ТА ЗАСОБІВ РЕАЛІЗАЦІЇ

### 2.1. Вибір операційної системи для розробки

Після детального аналізу додатка, що потрібно розробити, для його більш зручного та якісного виконання, потрібно обрати конкретні інструменти.

Перш за все – вибір операційної системи. На сьогоднішній день для розробки програмного забезпечення, особисто для мене, найкраще підходить Linux. Тому ОС для виконання роботи було обрано його дистрибутив Ubuntu 18.04 LTS. Оскільки ПЗ, яке розробляється, у вигляді веб-орієнтованого додатку, то проблем з кросплатформністю виникати не буде. Основними перевагами обраної ОС перед іншими є:

- Open Source;
- безпека користування (невелика кількість вірусів);
- висока швидкодія;
- невисоке споживання ресурсів.

Недоліками цієї системи являються:

- складність освоєння для початківців;
- відсутність деякого ПЗ;
- необхідність знання та роботи з консоллю;
- велика відмінність від інших ОС.

Ubuntu 18.04 LTS призначений для звичайного використання на комп'ютерах. Він дістав великої популярності серед студентів та програмістів. Це пов'язано з тим, що його засоби дозволяють зручно та швидко виконувати системні команди засобом командної строки та тим, що на ОС практично відсутні віруси.

Ubuntu досить молода операційна система і була випущена 2004 р. але розвивається досить стрімко. Кожні півроку випускається нова версія, а щоразу через два роки випускається стабільна версія, яка підтримується.[17]

## 2.2. Обґрунтування вибору мови програмування

Для розробки ПЗ необхідно також обрати мову програмування, за допомогою якої буде виконуватися клієнтська частина. Насправді вибір був серед скрипкових мов: VisualBasicScript і JavaScript. Особливістю даних мов є те, що:

- вони виконуються не на сервері а в самому браузері, саме тому вони і називаються клієнтськими;
- мають високу швидкодію в порівнянні з серверними мовами програмування;
- сильно залежать від браузера то його версії, тобто результат роботи цих мов буде відрізнятися та різних системах;
- інколи через відсутність підтримки даного інструмента на стороні клієнта взагалі може не виконуватися скрипт.

Через велику популярність та знання було обрано JavaScript. Це основний інструмент, який використовується для написання ПЗ, оскільки основною задачею є відображення в браузері даних про користувачів, та їхніх статистичних даних.

Синтаксис JavaScript було утворено на базі Сі та Java, але таким чином, що б спростити використання даної мови програмістам. Так сталося, що альтернатив клієнтському JS практично не існує.[18]

При навчанні JavaScript, як і будь-якої мови програмування, виникають труднощі, але загалом синтаксис досить простий для початківців. Це пов'язано з тим, що код на цій мові схожий до Java і Сі (Рис.5.)

```

1  function checkAge(age) {
2      if (age > 18) {
3          return true;
4      } else {
5          return confirm('Родители разрешили?');
6      }
7  }
8
9  var age = prompt('Ваш возраст?');
10
11 if (checkAge(age)) {
12     alert( 'Доступ разрешен' );
13 } else {
14     alert( 'В доступе отказано' );
15 }

```

Рис.5. Приклад синтаксису мови JavaScript

Клієнтський JavaScript, як було сказано вище, скрипкова мова програмування, тому не потребує компілювання. Це є як перевагою так і недоліком, адже послідовне виконання коду забезпечує його швидкодію, але не забезпечує високої надійності. При виникненні помилки, код просто не буде далі виконуватися і програма зупиниться на моменті виникнення помилки, тоді як в не скрипкових мовах програмування помилка була б виявлена на етапі компілювання.

Існують засоби розробки, які в реальному часі сповіщають про лексичні та синтаксичні помилки, але це лише частина з можливих помилок. JavaScript багато хто вважає поганим інструментом розробки через деякі нелогічні та незрозумілі програмістам можливості. Часто це приводить до прихованих помилок, які дуже важко відловити розробнику, адже ця унікальність мови програмування вводить в оману. Також існують труднощі відлагоджування коду, для цього потрібно використовувати модулі доповнення (плагіни) веб-орієнтованого розробника. Особливо виникають проблеми у початківців, які перейшли на JavaScript з мови більш низького рівня програмування, тому що там існує строга типізація та необхідність компілювати програму.

JavaScript можна використовувати в поєднанні з багатьма мовами програмування і скрипти даної мови можна інтегрувати в будь-яку веб-сторінку незалежно від розширення файлу. Наприклад, JavaScript може використовуватися всередині скриптів, написаних на PHP чи Perl. Це робить мову програмування більш привабливою для програмістів і більш гнучкою у використанні.

Мало того що JavaScript досить швидко виконується на стороні клієнта, так він ще й зменшує навантаження на сервер, тим самим збільшуючи швидкодію всієї системи.

Особливістю JavaScript є те, що він являється не типізованим, а це, в свою чергу, дає більше можливостей для програміста, адже не потрібно турбуватися про приведення типів даних, яке відбувається автоматично. Також, не потрібно писати декілька однотипних функцій для різних типів, адже одна функція може працювати з будь-яким типом даних.

Ще однією перевагою даного інструменту є те, що в ньому реалізовано як об'єктно-орієнтоване програмування так і функціональне, програміст на свій розсуд може обрати бажаний йому стиль.

Основні проблеми заключаються в тому, що різні браузері виконують код, написаний на JavaScript, по-різному. Це робить деякі незручності при розробці, адже необхідно перевіряти виконання програми в кожному з браузерів. Часто відслідковувати та вирішувати такі проблеми непросто, адже браузерів досить багато і під кожен потрібно підлаштовуватися.

Також ця мова програмування не стоїть на місці і активно розвивається, за останні роки було випущено нові специфікації ECMAScript(Рис.6.). Але знову ж таки, оскільки ця мова програмування не компілюється, то виникає проблема підтримки нових версій різними браузерами.

| Видання | Ім'я   | Дата випуску         | Зміни від попереднього видання   | Редактор                         |
|---------|--|----------------------|--|----------------------------------|
| 1       | ECMAScript 1                                       | Червень 1997 року    | -  | Гай Л. Стіл-молодший             |
| 2       | ECMAScript 2                                       | Червень 1998 року    | Зміни в редакції з метою забезпечення відповідності специфікації повністю міжнародному стандарту ISO / IEC 16262   | Майк Коуплішоу                   |
| 3       | ECMAScript 3                                       | Грудень 1999 року    | Додано регулярні вирази, краще керування рядками, нові оператори керування, обробка виключень try / catch, жорсткіше визначення помилок, форматування для числового виводу та інші покращення  | Майк Коуплішоу                   |
| 4       | ECMAScript 4                                       | Ніколи не випускався | Четверте видання було залишено через політичні розбіжності щодо складності мови. Багато функцій, запропонованих для Четвертого видання, були повністю знищені; деякі з них були включені до шостого видання.                                     |                                  |
| 5       | ECMAScript 5<br><a href="#">докладніше про ES5</a> | Грудень 2009 року    | Додано "строгий режим".<br>Додана підтримка JSON. Додано String.trim (). Додано Array.isArray (). Додані методи ітерації масивів.  | Пратап Лакшман, Аллен Вірфс-Брок |
| 5.1     | ECMAScript 5.1                                     | Червень 2011 року    | Це видання 5.1 стандарту ECMAScript повністю узгоджується з третім виданням міжнародного стандарту ISO / IEC 16262: 2011.  | Пратап Лакшман, Аллен Вірфс-Брок |
| 6       | ECMAScript 6<br><a href="#">докладніше про ES6</a> | Червень 2015 року    | Додано let і const.<br>Додано значення параметрів за замовчуванням. Додано Array.find (). Додано Array.findIndex ().   | Аллен Вірфс-Брок                 |
| 7       | ECMAScript 7                                       | Червень 2016 року    | Основні стандартні функції мови включають в себе блок-область видимості змінних і функцій, знищення моделі (змінні), відповідні виклики хвоста, оператор зведення в ступені ** для чисел await, асинхронні слова для асинхронного програмування. | Брайан Терлсон                   |
| 8       | ECMAScript 8                                       | Червень 2017 року    | Додано додавання рядків.<br>Додані нові властивості об'єкта. Додані функції Async. Додана спільна пам'ять.   | Брайан Терлсон                   |
| 9       | ECMAScript 9                                       | Червень 2018 року    | Додані властивості відпочинку / поширення.<br>Додано асинхронну ітерацію. Додано Promise.finally (). Додовнення до RegExp.   | Брайан Терлсон                   |

Рис. 6. Різні версії винання ES

Для уникнення цього, існують інструменти такі як Babel – це вид компілятора вхідними даними якого є код на мові програмування JavaScript однієї специфікації, а вихідними – код на тій же мові але іншої версії, яку розуміють браузер. Тобто, за допомогою цього компілятора можна писати код на сучасних версіях JavaScript, які мають кращий функціонал та більш зручні для програміста, а потім переводити його в більш старі версії, які підтримуються всіма браузерами. І тепер можна не задумуватись про проблему підтримки браузера.

В JavaScript використовується AJAX для асинхронних запитів на веб-сервер, це дозволяє в реальному режимі часу відображати дані на сторінці та надсилати їх на сервер без перезапуску сайту. Для відправки даних



використовується HTTP Request, а у JSON форматі приймаються відповідь від сервера (Рис.7.).

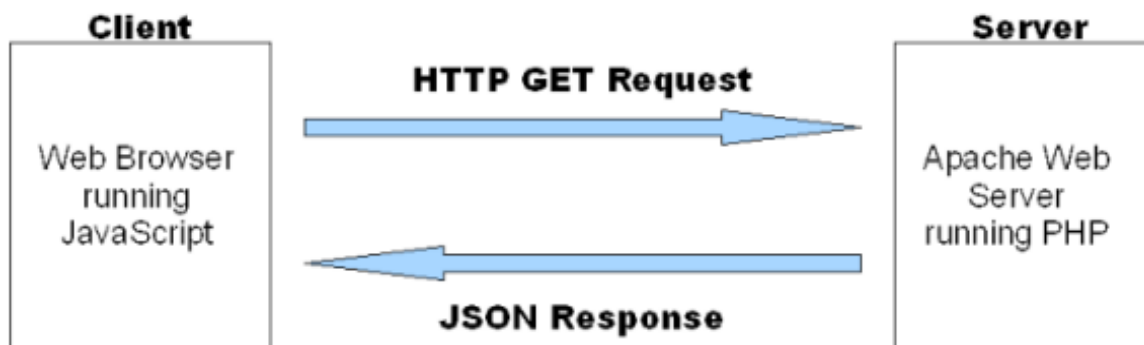


Рис.7. Схема обміну даними JavaScript з сервером

JavaScript користується великою популярністю на ринку праці, це також є перевагою, адже існує велика кількість літератури статей та документації, які допомагають вивчати та вирішувати задачі на цій мові програмування.

## 2.3. Сторонні засоби для покращення розробки

Часто для ще кращої розробки веб-додатків використовують сторонні бібліотеки та фреймворки. Я також не відмовився від такої можливості.

### 2.3.1. Вибір основної бібліотеки React

Серед можливих варіантів вибору бібліотек і фреймворків було розглянуто найпопулярніші. Тому обирати приходилось із двох, а саме: AngularJS і ReactJS.

AngularJS – веб-фреймворк який перебуває у вільному доступі для використання програмістами та компаніями. Він базується на основі MVC(Рис.8.) шаблону, що дозволяє програмісту розподіляти код для відображення на сторінці та «бізнес» логіки програми. [19]

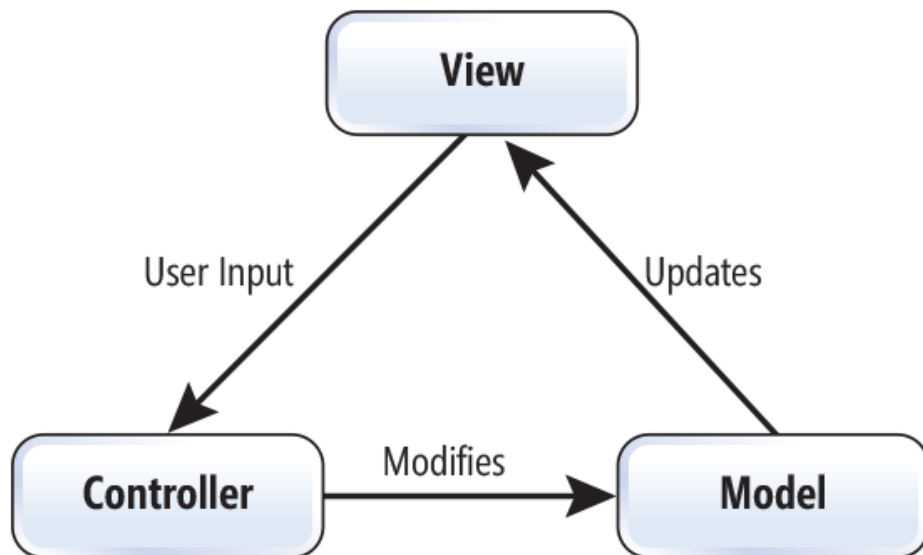


Рис.8. Структура MVC шаблону

MVC реалізовує розділення веб-додатків на три частини:

- інтерфейс користувача;
- керування (логіка);
- модель даних.

Використовується для того, щоб кожен з компонентів MVC був максимально незалежним один від одного. Це забезпечує гнучкість та простоту в реалізації великих проектів для яких найчастіше MVC і застосовується.[4]

AngularJS – дуже потужний інструмент, який багато переваг:

- двостороння прив’язка даних – при зміні даних в моделі дана будуть змінюватися в відображенні і навпаки;
- використання власно написаних директив;
- структура MVC;
- можливість використання фільтрів для бажаного відображенні даних в заданому форматі.

Недоліки :

- специфічний стиль написання коду;
- складна структура проекту;

- складний обмін даними між компонентами (import, output).

React як і AngularJS доступна для вільного користування бібліотека для реалізації користувацьких інтерфейсів. Оскільки в моделі MVC за допомогою React реалізується тільки користувацький інтерфейс. Також основною особливістю даної бібліотеки є те, що вона дозволяє розроблювати одно сторінкові веб-додатки. Також React вирішує проблему часткового оновлення контенту сторінки, оскільки він має інструмент як віртуальний DOM. За його допомогою можна не оновлювати кожного разу сторінку повністю, а тільки ту частину, яка цього потребує.[20]

В основі React використання components, virtual DOM, state і props(Рис.9.). Компонент – це структурна одиниці додатків, написаних за допомогою цієї бібліотеки. virtual DOM – це аналог звичайного DOM, який використовується для перерисовки відображення сторінки. State – використовується для збереження поточного стану компонента з можливістю подальшого їх змінення. В залежності від змін відповідно буде змінюватися і сам вигляд сайту. Props – це дані які передаються з батьківського компонента.

Бібліотека дозволяє якісно будувати графіки та діаграми.

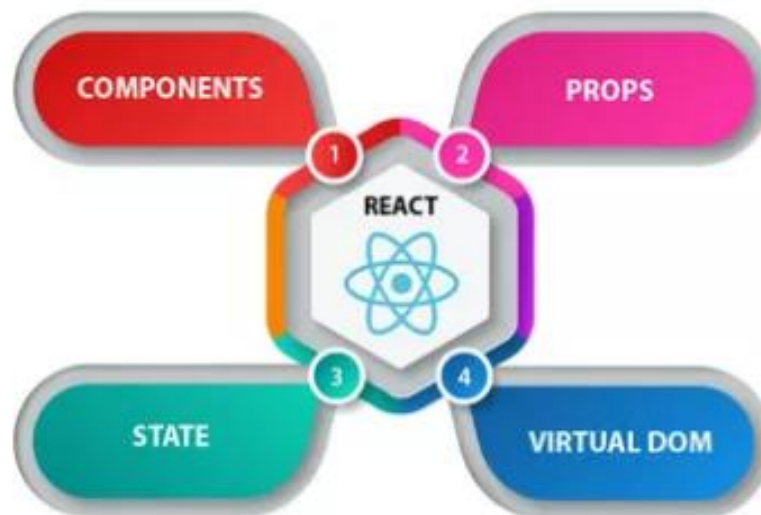


Рис.9. Основи основ бібліотеки React

Переваги React:

- висока швидкодія;
- віртуальний DOM;
- використання життєвого циклу компонента(Рис.10.);
- простота написання коду;
- швидкість розробки;
- використання бібліотеки з будь-яким фреймворком.

Недоліки:

- невелика API;
- React – це тільки інтерфейс.

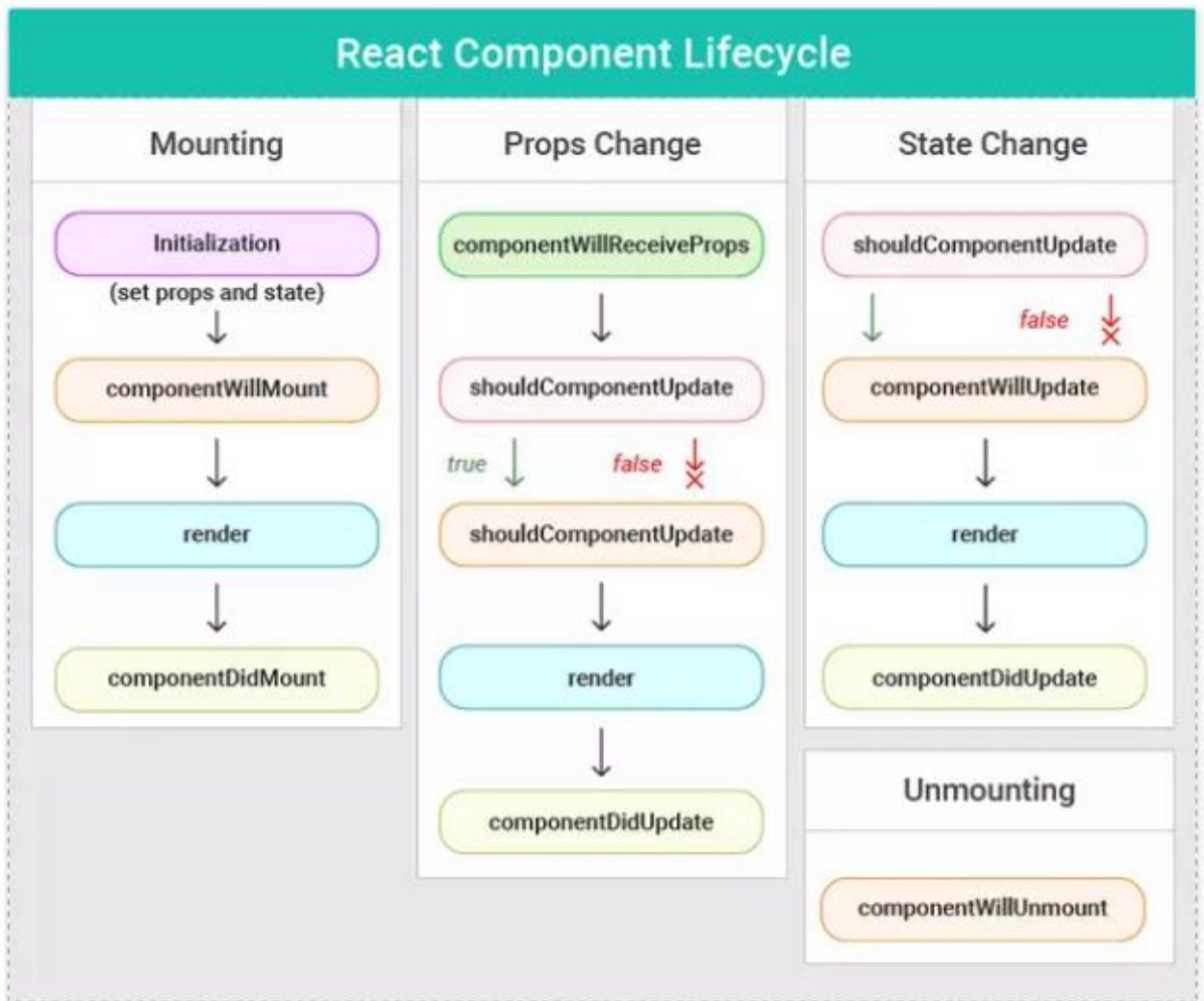


Рис.10. Життєвий цикл React компонента

Порівнявши бібліотеку та фреймворк для виконання дипломного проекту було обрано саме React. Це пов'язано з тим, за допомогою цього інструменту можна легко та якісно розробити програмне забезпечення, яке задовольнить всім вимогам, що було поставлено раніше.

### 2.3.2. Технологія JSX

Разом з React зручно використовувати JSX, хоча можна обходитися і без нього(Рис.11.).

```
ReactDOM.render(  
  
  React.DOM.div(  
  
    { id: 'test' },  
  
    React.DOM.h1(null, 'A title'),  
  
    React.DOM.p(null, 'A paragraph')  
  
  ),  
  
  document.getElementById('myapp')  
  
)
```

Рис.11. Приклад маніпуляцій з HTML без використання JSX

Ця технологія яка виглядає як суміш HTML CSS і JavaScript(Рис.12.).

Насправді таке незвичне поєднання – це простий JavaScript код, тільки в іншому вигляді. Як було сказано раніше, що будь-який код з використанням JSX можна написати за допомогою звичайного JavaScript. Це твердження обумовлене тим, що JSX транспонується в JavaScript за допомогою Babel.

```

ReactDOM.render (

  <div id="test">

    <h1>A title</h1>

    <p>A paragraph</p>

  </div>,

  document.getElementById( 'myapp' )

)

```

Рис.12. Приклад що реалізовує ті ж дії що зображено на Рис.11. з використанням JSX

JSX дозволяє наглядно працювати з розміткою та наповненням сайту. Наприклад, існує можливість присвоєння змінній HTML виразу, і потім багаторазово використовувати цю змінну у відображенні.

Але синтаксис JSX трохи відрізняється від звичайного HTML, адже він по своїй природі ближчий до JavaScript. Що б не виникало конфліктів при транспонуванні, назва атрибуту в HTML class в JSX було замінено на className. Адже слово «class» є зарезервованим в JavaScript. Так само не можна використовувати атрибути, які пишуться через дефіс.

### 2.3.3. Вибір інструменту для написання стилів

Для написання стилів можна використовувати також різні засоби. Сьогодні розробники веб-інтерфейсів найчастіше використовують CSS, Less і Scss (Sass). Вибір було зроблено на користь останнього.

Код Scss браузер не розуміє, тому його потрібно компілювати в CSS(Рис.13.).

|    |      |          |       |      |                           |            |
|----|------|----------|-------|------|---------------------------|------------|
|    |      |          |       |      | <b>ІАЛЦ.045440.004 ПЗ</b> | Лист<br>26 |
| Зм | Лист | № докум. | Підп. | Дата |                           |            |

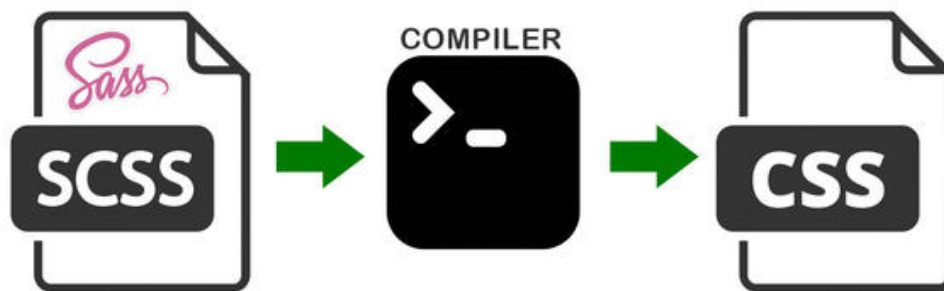


Рис.13. Схема перетворення Scss в CSS

Переваги Scss:

- сумісний з усіма версіями CSS;
- забезпечення використання констант;
- імпортування шаблонів
- містить менший об'єм коду в порівнянні з CSS;
- швидке написання стилів;
- можливість написання вкладених стилів;
- використання математичних обчислень.

Недоліки:

- не кожен розробник знає синтаксис Scss, на початковому етапі, через це, можуть виникати проблеми;
- необхідність перетворення Scss в CSS.

## 2.4. Середовище розробки Visual Studio Code

Середовище в якому розробляється програма зазвичай обирається програмістом опираючись на власні уподобання. Останнім часом, великої популярності набув текстовий редактор коду Visual Studio Code.

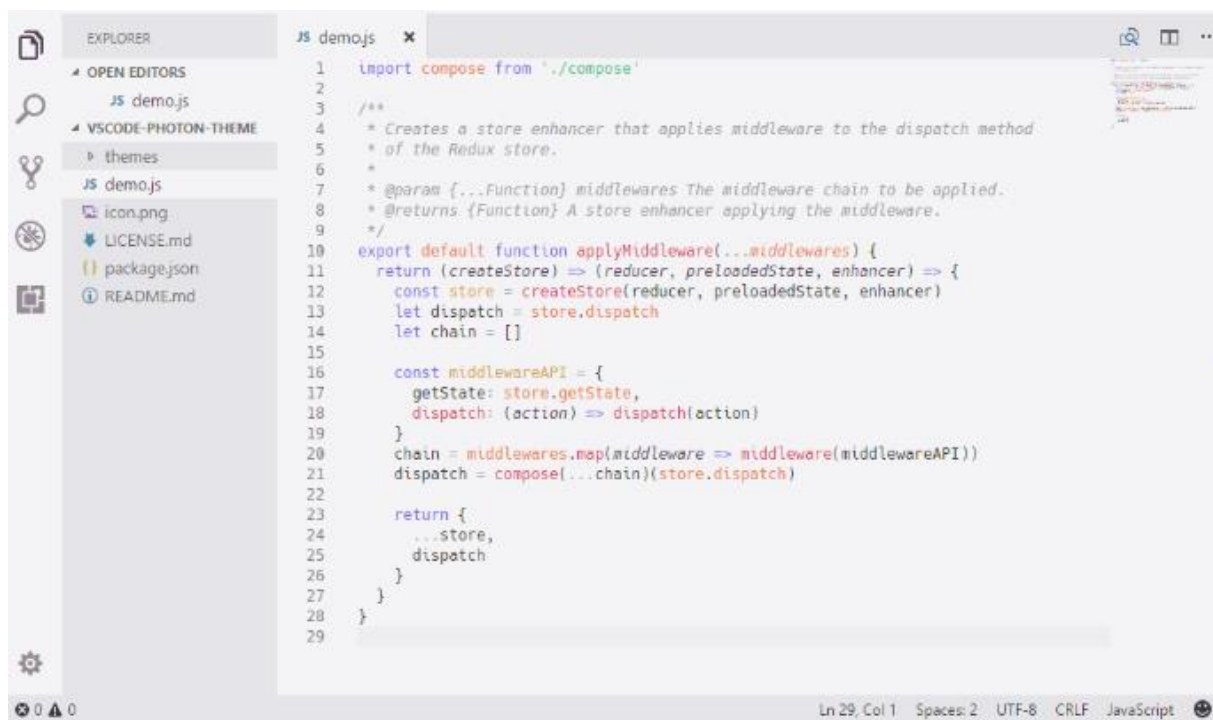


Рис.14. Графічний інтерфейс Visual Studio Code

#### Переваги VS Code:

- простота інтерфейсу(Рис.14.);
- влаштована командна рядок (термінал);
- система контролю версій Git;
- відлагоджувач;
- автоматичне редагування коду;
- можливість встановлення розширень;
- паралельне редагування коду в декількох файлах;
- можливість вибору різних тем;
- швидкодія;
- різноманітні підказки для програміста;
- відображення в реальному режимі часу веб-додатків
- безкоштовний.



Недоліки:

- необхідність налаштування
- встановлення розширень для кожної мови програмування

|    |      |          |       |      |                           |      |
|----|------|----------|-------|------|---------------------------|------|
|    |      |          |       |      | <b>ІАЛЦ.045440.004 ПЗ</b> | Лист |
| Зм | Лист | № докум. | Підп. | Дата |                           | 29   |

### Розділ 3. АРХІТЕКТУРА ТА ОПИС РЕАЛІЗАЦІЇ РОЗРОБЛЕНОГО ПЗ

Додаток було створено за допомогою команди create-react-app, що допомогло створити пустий шаблон React додатку. Це допомагає не встановлювати самостійно react та react-dom. Вони забезпечують можливість використання API React, який коротко було описано вище. Після чого було налаштовано конфігураційні файли для більш зручної розробки програмного забезпечення, це забезпечило автоматичний перезапуск додатка в браузері при змінах в коді.

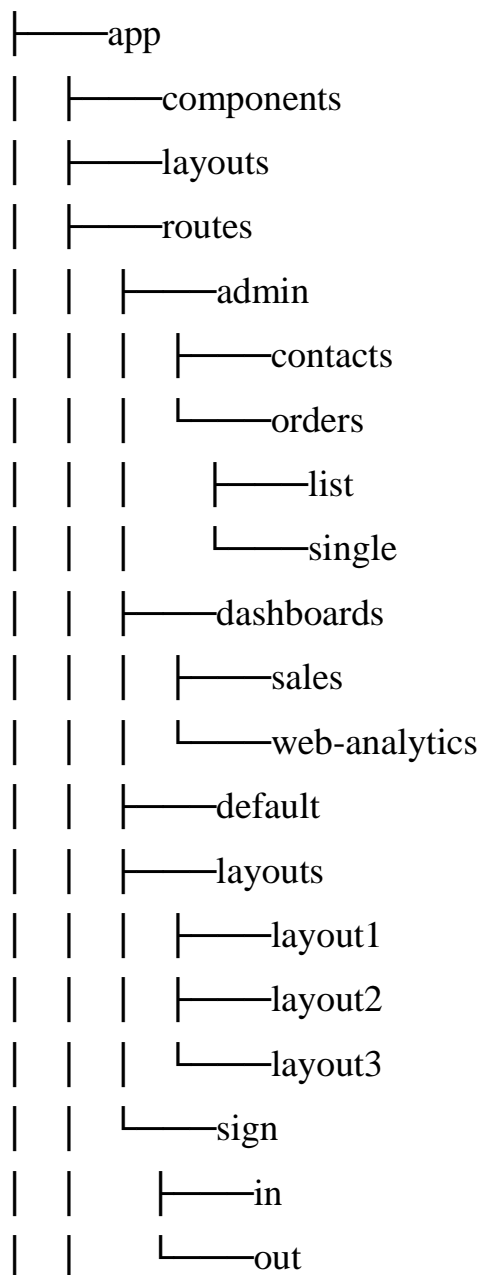
Перед початком роботи над програмним забезпеченням було сформовано приблизну архітектуру додатку. Поступово створюючи пусті компоненти для майбутнього наповнення їх вмістом, було отримано досить розгалужену та велику структуру додатку. Перш за все було створено основні компоненти для забезпечення розмітки сайту. До них входять: панель заголовку, навігації, та основного вмісту. Користувач тільки їх буде бачити на сторінці, тому все інше буде відображатися всередині основних компонентів.

Маючи її можна формувати систему маршрутизації, яка містить в собі основне наповнення додатку. Для забезпечення програми всім функціоналом було створено компоненти для веб-аналітики, входу в кабінет, відображення інформації про користувачів, продажів.

Для можливості зміни та вибору стилів навігації було впроваджено декілька шаблонів, які в подальшому можна розширювати та створювати нові.

Після виконання всього описаного було отримано наступну структуру директорій:

|           |             |                 |              |             |                           |                   |
|-----------|-------------|-----------------|--------------|-------------|---------------------------|-------------------|
|           |             |                 |              |             | <b>ІАЛЦ.045440.004 ПЗ</b> | <b>Лист</b><br>30 |
| <b>Зм</b> | <b>Лист</b> | <b>№ докум.</b> | <b>Підп.</b> | <b>Дата</b> |                           |                   |



Відобразити повністю структуру, яка буде включати в себе всі створені файли, складно, адже деякі директорії містять до десяти файлів.

Веб-додаток виконано у вигляді одно single-page application (SPA).

SPA – це додаток, що запускається та працює в браузері, який не виконує перезавантаження сторінки під час його застосування користувачем. Останнім часом цей вид веб-додатків набув великої популярності серед програмістів та замовників. Його використовують Facebook, Gmail, GitHub, тому більшість користувачів користуються односторінковими додатками щодня.

Його популярність обумовлена тим що в минулому браузері та JavaScript були повільними, адже для відображення нової сторінки сайту необхідно було кожного разу відсилати запити на сервер для отримання HTML. І після перезавантаження сторінки можна побачити інформацію. Проблеми виникали при відображенні на сторінці великої кількості даних, адже, при цьому, розміри HTML дуже великі і для передачі із серверу на клієнт витрачалось багато часу.

В SPA ця проблема вирішена, адже не додаток має тільки один HTML, який просто частково змінює своє наповнення. І, навіть, при невеликих затримках відображення інформації, користувач не буде бачити пустий екран браузера, а буде залишатися на попередній сторінці.

#### Переваги SPA:

- швидкість завантаження сторінки, адже HTML і CSS завантажуються тільки один раз, а при переході на нову сторінку змінюється тільки їхній вміст;
- простота розробки, адже не потрібно на сервері писати код для відображення відповідної сторінки;
- простота відлагоджування, оскільки без перезавантаження зберігаються всі мережеві запити та операції. Це допомагає краще досліджувати додаток;
- робота з мобільними версіями сайту також спрощується. Існує можливість використовувати один і той же код як для мобільного так і для повноцінного веб-додатка;
- ефективність використання кешу. В SPA для збереження деяких даних відправляє тільки один запит на сервер, а згодом ці дані можуть використовуватися в будь-якому місці сайту.

Мінуси:

- довготривале початкове завантаження. Цю проблему виправлено в сучасних бібліотеках та фреймворках;
- втрата пам'яті може призвести до затримок складних додатків.

Бібліотека React забезпечує можливість написання односторінкових додатків.

Для реалізації SPA використовується технологія маршрутизації, адже React - це бібліотека а не фреймворк і він не містить в собі початкових засобів реалізації маршрутизації. React Router використовує вище згаданий JSX.

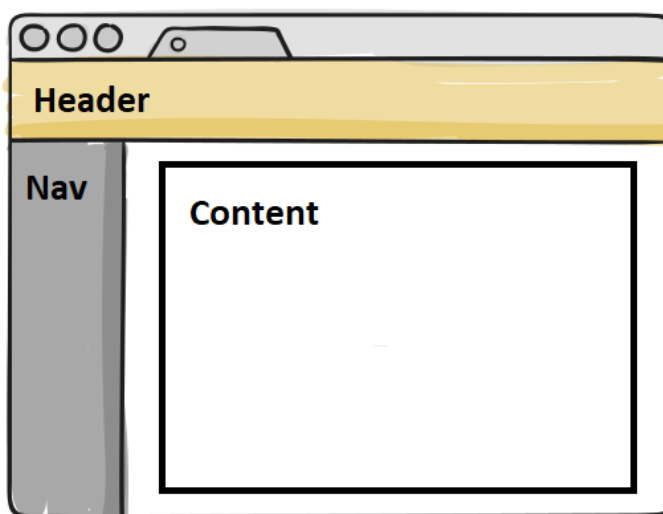


Рис.15. Структура відображення веб-додатка

React Router забезпечує відображення конкретного контенту в залежності від дій користувача. Тобто, веб-додаток може не змінюватися повністю, зміни будуть відбуватися лише в певному місці. В реалізованому програмному забезпеченні зміни відображення інформації (HTML) відбуваються лише в області Content (Рис.15.). Інші блоки програми мають несуттєві зміни при користуванні, але ці зміни забезпечуються не маршрутизацією, а змінами слів.

При маршруті «**/admin/contacts**» відображається список користувачів. Список просто вставляється в HTML Content а Header і Nav не змінюються.(Рис.16.).

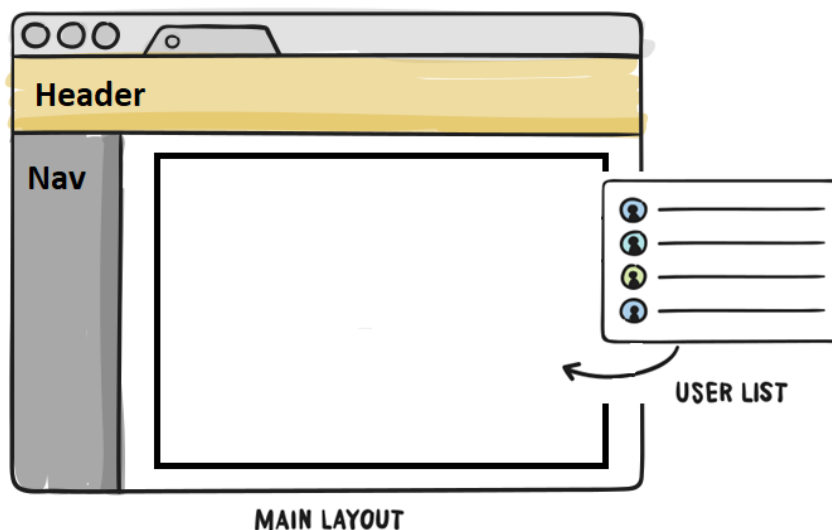


Рис.16. Схема відображення маршрутизації для списку користувачів

За допомогою аналогічного виконання було виконано систему маршрутизації веб-додатку. Система була розроблена таким чином, щоб розділити по різних маршрутах різне відображення інформації, але при цьому не ускладнюючи глибокою вкладеністю маршрутів. Це забезпечить простоту та комфорт використання додатку.

Для наочності необхідно навести перелік всіх маршрутів:

- / - домашня сторінка;
- /dashboards/sales – відображення продаж
- /dashboards/web-analytics – відображення аналітики
- /admin/contacts – відображення інформації про користувачів
- /admin/orders – відображення списку замовлень
- /sign/in – вхід в кабінет

В загальному в залежності від маршруту відображення сторінки буде виглядати як на Рис.17.

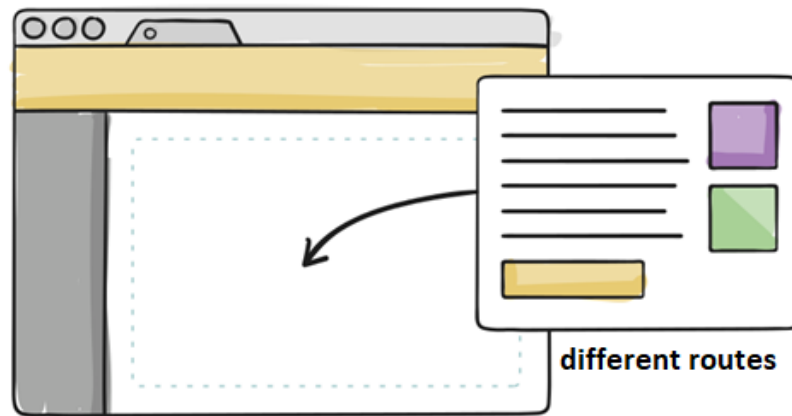


Рис.17. Загальна схема відображення вмісту в залежності від різних маршрутів.

Для зручності, маршрутизація відповідає архітектурі директорій, підібрані відповідні назви для директорій, файлів, та маршрутизації. При цьому виникає зв'язок між назвою маршруту та розміщенням відповідного коду.

Тепер можна більш детально описати саму реалізацію. Почнемо з основних компонентів: панелі навігації та заголовку.

Компонент називається `AppLayout`. В ньому за допомогою `JSX` розміщуються статичний заголовок, реалізований за допомогою звичайного тега `<header></header>`.

Навігація по сторінці відображається вже власно написаним тегом `<SideNav/>`. Цей вміст тега незмінний, але існує функція вибору різних тем. Різниця проявляється саме на відображенні навігації. Було виконано три різні шаблони, які відрізняються розміщенням та стилями. Це було виконано для того, що б користувач міг самостійно обрати бажану тему, яка йому буде найбільш до вподоби.

По замовчуванню встановлено стиль, який показано на Рис.18.

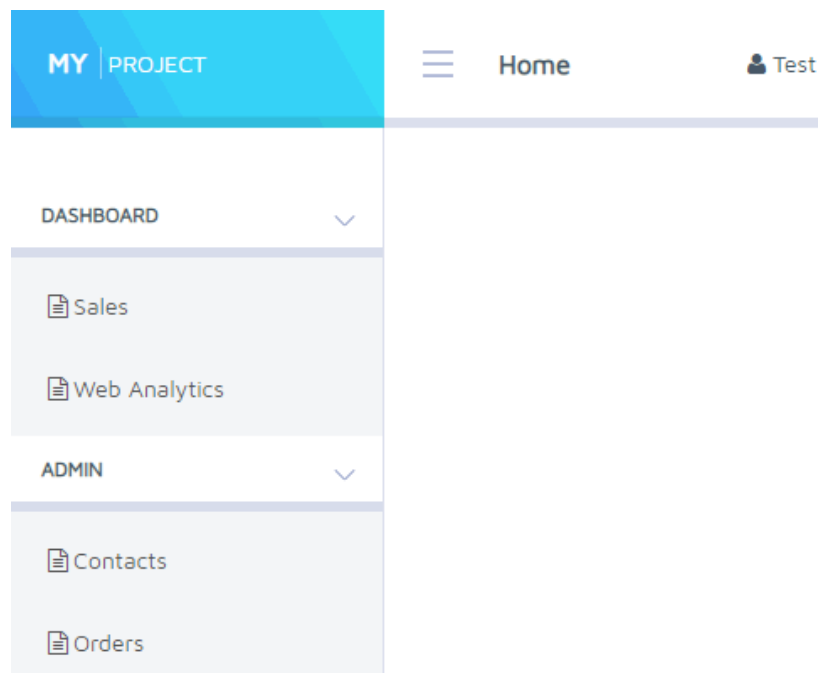


Рис.18. Стиль відображення панелі навігації що встановлена за замовчуванням

При бажанні користувач може встановити інші стилі навігації (Рис.19., Рис.20.).

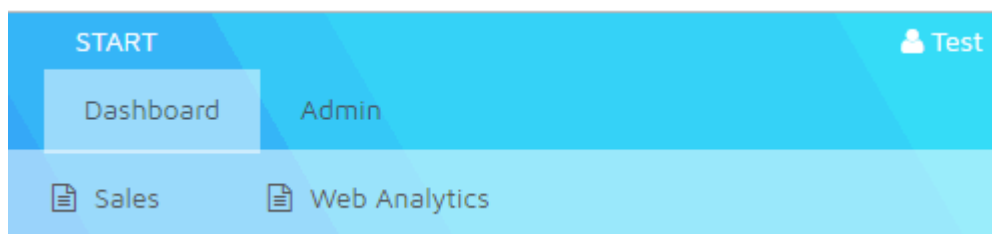


Рис.19. Стиль відображення панелі навігації

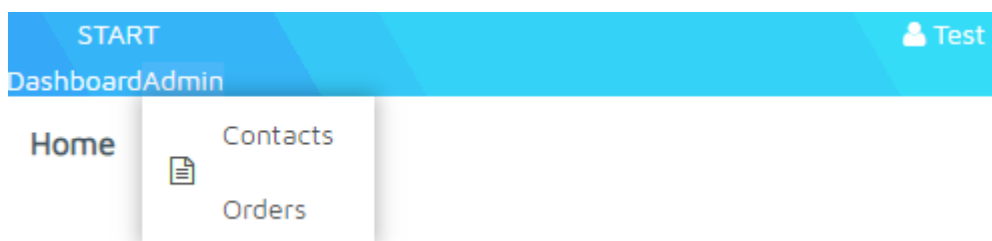


Рис.20. Стиль відображення панелі навігації

Ця функція була виконана шляхом звичайної заміни розмітки та стилів.



Основний вміст сторінки може мати п'ять станів:

- початкова домашня сторінка;
- продажі;
- веб-аналітика;
- контакти;
- замовлення.

Початкова сторінка відображається користувачу при запуску програми. Ця сторінка реалізована досить просто але несе в собі необхідну інформацію для правильного використання веб-додатку. Це свого роду інструкція для початківців.

Сторінка продаж вже складніша і несе в собі корисну інформацію для адміністраторів сайту.

Компонент використовується під назвою тега <Sales/>, що дозволяє без труднощів за значенням ідентифікувати його.

Цей компонент складається з:

- TopCustomers
  - customer
  - revenue
- KeyMetrics
  - metric
  - target
  - actual
- RevenueQTD
  - region
  - actual
  - target
- RevenueYTD
  - region

- units
- target
- actual
- ProductSalesYTD
  - region
  - units
  - target
  - actual
- MarkedShare
  - company
  - totalMarket

Виконано у вигляді окремих діаграм, які розміщуються на одній сторінці. Було використано бібліотеку Chart з відповідним тегом <Chart/>. Для заголовків у діаграмі застосовувався тег <Text/>, згодом для кожної діаграми за допомогою циклів відображається кожний рядок з необхідними полями. Це забезпечує малий об'єм коду та можливість виводити будь яку кількість рядків.

Відмінність в реалізації між кожною діаграмою невелика. Кожна з них має різні заголовки та деякі відмінності в наповненні. Також в циклі відображення використовуються різні масиви об'єктів. Це унеможливило написання одного компонента для реалізації кожної з діаграм.

TopCustomers – відображає топ клієнтів за доходами.

KeyMetrix – ключові показники з початку року.

RevenueQTD – дохід за останній квартал.

RevenueYTD – дохід з початку року.

ProductSalesYTD – продажі продукції від початку року.

MarkedShare – відображення ринкової частки продаж компанії в порівнянні з іншими.

Сторінка веб-аналітики включає в себе:

|    |      |          |       |      |                           |      |
|----|------|----------|-------|------|---------------------------|------|
|    |      |          |       |      | <b>ІАЛЦ.045440.004 ПЗ</b> | Лист |
| Зм | Лист | № докум. | Підп. | Дата |                           | 38   |

- TopKpis
  - sessions
  - users
  - pagesViews
  - pages
  - sessionDuration
  - period
  - bounceRate
- Trend
- Details
  - referral
  - country
  - browser
  - city

TopKpis – ключові показники продуктивності(Рис.20.). Користувач може обрати будь-який з них і подивитися динаміку змін за період часу Trends.

Для його реалізації використовується SVG та Chart. Вони забезпечують відображення невеликого графіка за певний період під кожним елементом ключових показників. Це забезпечує можливість одночасного перегляду графіків для кожного з параметрів продуктивності.

Trends – графік за певний період ключових показників продуктивності(Рис.20.). Для більш детального перегляду невеликих графіків цей компонент реалізує відображення великого графіка одного з показників продуктивності при натисненні на нього. Для побудови графіка використовувалася бібліотека Chart.

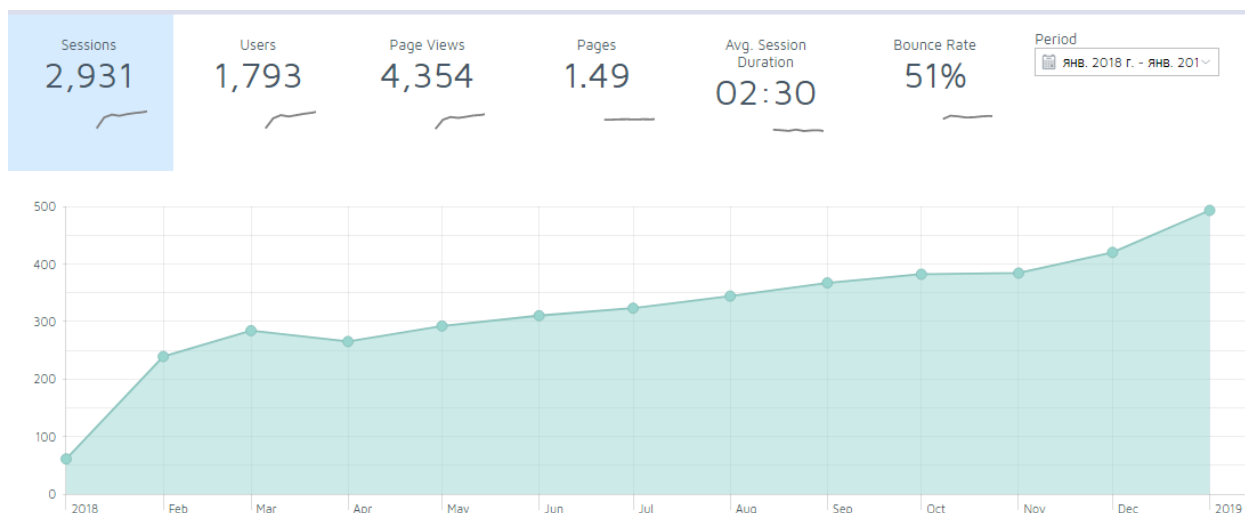


Рис.20. Відображення на сторінці TopKpis i Trends

Details – дані за країною браузером містом направлення(Рис.21.). При натисненні на один з пунктів можна подивитися більш детально про його параметри.

Для реалізації використовується одна функція, яка приймає параметром один з підрозділів у вигляді об'єкта. За допомогою інкапсуляції в JSX відображаються дані на сторінці. Кожний рядок підрозділу Details виводиться в циклі в тілі якого виводяться назва, кількість та невелика діаграма яка відповідає числовому значенню кількості.

| Referral              | Sessions | Country                              | Sessions | Browser                  | Sessions | City          | Sessions |
|-----------------------|----------|--------------------------------------|----------|--------------------------|----------|---------------|----------|
| www.et.com            | 424      | Guam                                 | 170      | Android/Safari           | 195      | Leraton       | 65       |
| www.odit.com          | 192      | Kenya                                | 165      | iOS/Firefox              | 190      | New Napoleon  | 58       |
| www.enim.com          | 191      | New Caledonia                        | 160      | Mac OS/Internet Explorer | 181      | East Ava      | 57       |
| www.voluptas.com      | 180      | United States Minor Outlying Islands | 155      | iOS/Safari               | 181      | Jerrellmouth  | 56       |
| www.commodi.com       | 179      | Mali                                 | 119      | Ubuntu/Internet Explorer | 180      | South Earnest | 54       |
| www.non.com           | 147      | Sierra Leone                         | 107      | Android/Firefox          | 178      | McDermottfurt | 53       |
| www.reprehenderit.com | 143      | Cocos (Keeling) Islands              | 106      | iOS/Internet Explorer    | 177      | New Blaise    | 53       |
| www.perferendis.com   | 126      | Barbados                             | 104      | Android/Opera            | 175      | New Jovani    | 53       |
| www.maiores.com       | 114      | Hong Kong                            | 103      | Mac OS/Firefox           | 174      | Anastacioview | 53       |
| www.corrupti.com      | 104      | Samoa                                | 101      | Ubuntu/Safari            | 171      | Cheyenneport  | 52       |
| www.id.com            | 102      | Comoros                              | 101      | Mac OS/Opera             | 168      | Vernview      | 52       |
| www.quibusdam.com     | 102      | Belarus                              | 99       | Ubuntu/Opera             | 162      | Hayleechester | 52       |

Рис.21. Відображення на сторінці Details

Сторінка контактів складається зі списку всіх користувачів, поля пошуку користувача(Рис.22).

Пошук реалізовано через TextField, який двонаправленою прив'язкою даних зв'язує текстове поле та змінну по якій відбувається пошук та фільтрація.

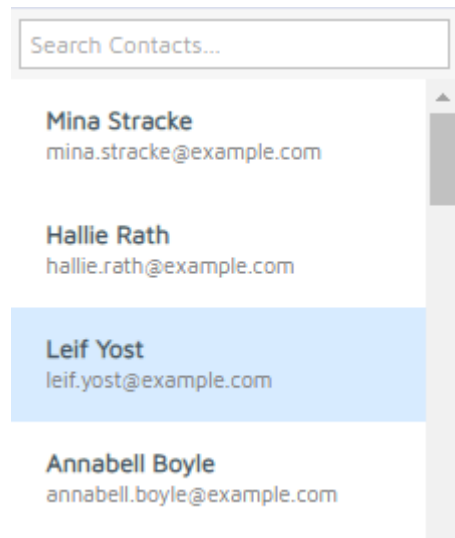


Рис.22. Відображення списку користувачів та їх пошуку

При виборі користувача відображаються дані про нього, з можливістю адміністратора редагувати та заповнювати поля(Рис.23.):

- firstName
- lastName
- primaryEmail
- secondaryEmail
- mobilePhone
- officePhone
- homePhone
- street
- nuberHome
- country
- region

## Leif Yost

First Name

Leif

Last Name

Yost

Email

Primary

leif.yost@example.com

Secondary

Phone

Mobile

Office

Home

Address

Street

Leda Oval

No.

78

Zip

21947

City

New Lilyan

Country

Grenada

State

Рис.23. Відображення даних про користувача

Сторінка замовлень складається зі списку замовлень та секції фільтрів.

Фільтрувати можна за номером замовлення, датою, країною, містом, та клієнтом що його здійснив, це надає можливість якісного та швидкого пошуку замовлення(Рис.24.).

|                      |                      |                      |
|----------------------|----------------------|----------------------|
| Order No.            | Date                 | Customer             |
| <input type="text"/> | <input type="text"/> | <input type="text"/> |
| Country              | City                 | Total                |
| <input type="text"/> | <input type="text"/> |                      |

Рис.24. Секція фільтрування замовлень

Дані які відображаються в списку замовлень:

– orderNumber

- date
- customer
- country
- city
- total

Кожне замовлення можна відкрити для отримання більш детальної інформації(Рис.25.). В розширеному форматі додаються поля:

- email
- status
- notes
- productList
  - productName
  - count
  - discount
  - Price
  - totalPrice
  - totalDiscount
  - tax
  - totalPriceWithTax

|    |      |          |       |      |                           |      |
|----|------|----------|-------|------|---------------------------|------|
|    |      |          |       |      | <b>ІАЛЦ.045440.004 ПЗ</b> | Лист |
| Зм | Лист | № докум. | Підп. | Дата |                           | 43   |

|              |              |                 |                            |              |
|--------------|--------------|-----------------|----------------------------|--------------|
| <b>Order</b> |              | <b>Customer</b> |                            | <b>Extra</b> |
| Order No.    | 10000        | Customer        | Derek Connelly             | Notes        |
| Date         | Dec 31, 2009 | Email           | derek.connelly@example.cor |              |
| Status       |              | Country         | Gibraltar                  |              |
|              |              | City            | Haagfort                   |              |

| Product      | Qty | Discount | Unit Price | Regular     | Discount   | Tax               | Total       |   |
|--------------|-----|----------|------------|-------------|------------|-------------------|-------------|---|
| Reilly 2     | 90  | 5%       | \$847.29   | \$76,256.10 | \$3,812.81 | \$10,866.49 (15%) | \$83,309.78 | ✖ |
| Schuppe 7    | 83  | 5%       | \$240.68   | \$19,976.44 | \$998.82   | \$3,415.97 (18%)  | \$22,393.59 | ✖ |
| Davis 5      | 2   | 5%       | \$257.50   | \$515.00    | \$25.75    | \$68.5 (14%)      | \$557.75    | ✖ |
| Carroll 9    | 90  | 5%       | \$347.28   | \$31,255.20 | \$1,562.76 | \$4,750.79 (16%)  | \$34,443.23 | ✖ |
| McLaughlin 7 | 71  | 5%       | \$428.08   | \$30,393.68 | \$1,519.68 | \$3,464.88 (12%)  | \$32,338.88 | ✖ |
| Kuvalis 4    | 26  | 5%       | \$79.26    | \$2,060.76  | \$103.04   | \$274.08 (14%)    | \$2,231.80  | ✖ |
| Hodkiewicz 8 | 76  | 5%       | \$813.12   | \$61,797.12 | \$3,089.86 | \$11,741.45 (20%) | \$70,448.71 | ✖ |

Рис.25. Відображення інформації про замовлення

Доступ до бази даних існуючого сайту отримати складно, тому було прийнято рішення використовувати Firebase. Він надає бекенд сервіс для отримання різноманітних даних.

Firebase Authentication використовується для реєстрації та аутентифікації користувачів. Існує можливість збереження інформації про зареєстрованих користувачів.



## Розділ 4. РЕКОМЕНДАЦІЇ ДЛЯ ПОДАЛЬШОГО ВДОСКОНАЛЕННЯ

Розвиток та вдосконалення програмного забезпечення може ще більше спростити роботу адміністраторів.

Основними напрямками розвитку можна виділити:

- адаптація
- автоматизація розсилки email повідомлень користувачам
- можливість повернення коштів при помилковому замовленні
- створення сторінки відгуків та побажань користувачів
- підтримка передових технологій

Використання Інтернет ресурсів може бути здійснено різними типами пристроїв. З появою великої кількості смартфонів та планшетів частка використання Інтернету комп'ютерами різко зменшилася. На даний момент вона складає 35.25%(Рис.26.).

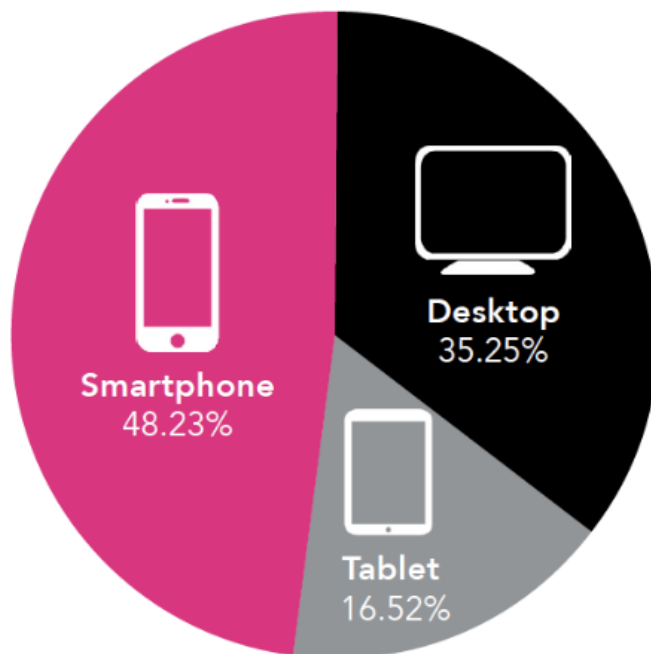


Рис.26. Частка використання Інтернету різними пристроями

Це говорить про те, що необхідно веб-додатки створювати адаптивними для різних типів пристрою, в залежності від розміру дисплею. Це дозволить використовувати всі функції розробленої програми будь-яким девайсам, що можуть відкрити веб-додаток.

Потрібно враховувати не тільки від пристрою а і його параметри.

Параметри розміру дисплею телефона дуже різні(Рис.26.). Для кожного з них потрібно забезпечити бажане відтворення веб-сторінки, це реалізується шляхом написання стилів для різного розміру екрана.

Аналогічна проблема присутня для планшетів та комп'ютерів.

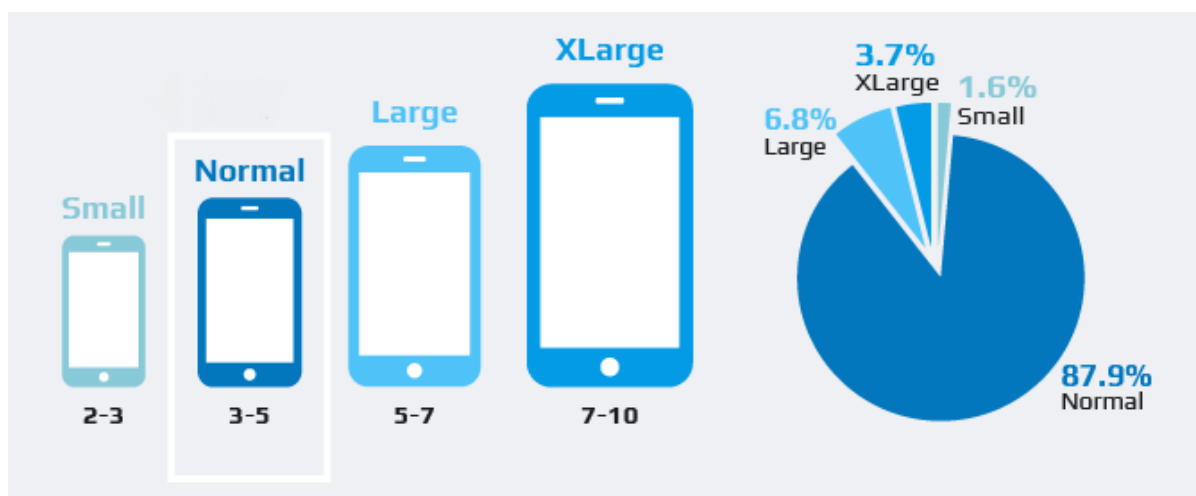


Рис. 26. Частки користувачів за розміром екрану

Також існує проблема коректного відображення додатка в різних браузерах. Між ними існують суттєві відмінності, особливо вирізняються Safari та Internet Explorer, саме вони часто спотворюють відображення сторінки. Проблема кросбраузерності не нова, але її потрібно вирішувати таким чином, що б веб-додатком можна було користуватися всіма сучасними браузерами.

Можна вдосконалювати та розширювати функціонал розробленої програми. Наприклад, реалізувати функцію автоматизованої розсилки email повідомлень. Ця функція може використовуватися для нагадування оплати

замовленої продукції. Також можна сповіщати клієнтів про знижки та акції, нові можливості.

Без автоматизації розсилки повідомлень це необхідно виконувати власноруч. При великій кількості користувачів на це потрібно витратити багато часу. Інколи ця потреба постає в неробочий час, виконання якої стає неможливим без автоматизації.

Важливою може стати функція автоматичного повернення коштів при оплаті помилкового замовлення. Ручне повернення коштів може стати проблемою, але права покупців слід враховувати. Вони повинні мати змогу повертати товар з зазначений термін, навіть, якщо його оплата проводиться онлайн.

Також потрібно створювати засоби інтерактиву з користувачами для взаємодії та діалогу з ними. Ця функція дозволить зрозуміти що необхідно змінити або додати, що б користувачу було зручніше користуватися сервісом. Сучасна конкуренція не дозволяє зневажати думками клієнтів, і для їх збереження та приросту необхідно впроваджувати такі можливості.

Для реалізації можна використовувати живий зв'язок. Для малого та середнього бізнесу це є проблемою, адже для цього необхідно відкривати офіси, які потребують високовартісного обслуговування.

Іншим вирішенням реалізація мобільного зв'язку та листування. Це більш реально реалізувати, але знову ж таки, необхідно особисто відповідати на дзвінки та перерахувати велику кількість листів.

Для більшого інтерактиву з клієнтами на сторінці адміністратора можна відображати список побажань та відгуків, які з часом можуть бути впроваджені для покращення сервісу.

В зв'язку з тим, що веб-технології дуже швидко розвиваються, то для покращення реалізації необхідно підтримувати систему. Сюди входить вдосконалення інтерфейсу, використання покращених інструментів розробки веб-додатка. За останні п'ять років у веб-орієнтованій сфері відбулася

«революція», адже було створено велику кількість нових можливостей та інструментів для розробника. І для реалізації якісного продукту необхідно вдосконалювати написаний раніше код передовими технологіями.

|    |      |          |       |      |                           |      |
|----|------|----------|-------|------|---------------------------|------|
|    |      |          |       |      | <b>ІАЛЦ.045440.004 ПЗ</b> | Лист |
| Зм | Лист | № докум. | Підп. | Дата |                           | 48   |

## ВИСНОВОКИ

У першому розділі було розглянуто та проаналізовано обов'язки адміністраторів сайту. До них входять:

- захист доступу до даних
- зв'язок з користувачами сайту
- робота з статистичними даними
- розробка майбутнього розвитку проекту
- аналітика

Часто при вирішенні даних задач виникають проблеми, оскільки це лише частина завдань які виконує адміністратор. Покращення роботоспроможності та вирішення даних задач і є основою для написання програмного забезпечення.

Перш ніж приступати до написання коду було розглянуто вже існуючі приклади реалізації системи моніторингу користувачів для адміністраторів сайту. Ця інформація дала розуміння як покращити аналоги та зробити продукт більш практичним.

Основними недоліками розглянутих проектів є:

- погане відображення великих об'ємів інформації
- незрозумілий інтерфейс
- повільний пошук даних

Основними перевагами є:

- дизайн
- широкий спектр можливостей даних систем

Проаналізувавши плюси і мінуси аналогів було сформовано вимоги для програми, що розробляється:

1. Панель для аналітики
2. Простий та зручний інтерфейс

### 3. Виведення статистичних даних

### 4. Пошукова система

Результатом дипломного проекту є система для забезпечення відображення та аналізу даних про користувачів сайту для більш комфортної роботи адміністраторів сайту.

Проаналізувавши можливі варіанти вибору інструментів для виконання дипломного проекту було обрано найефективніші.

Основні критерії вибору інструментів розробки програмного продукту:

- сучасність;
- простота;
- забезпечення безкоштовного використання;
- швидкодія;
- зручний користувацький інтерфейс;
- популярність.

Опираючись на ці критерії було обрано:

- JavaScript – мова програмування для написання клієнтської частини програмного забезпечення;
- Ubuntu – операційна система;
- React – JavaScript бібліотека, для забезпечення реалізації відображення в системі MVC;
- JSX – технологія для поєднання HTML, CSS та Javascript;
- Visual Studio Code – редактор коду для створення веб-додатків.

Веб-додаток реалізовано за допомогою технології SPA, що надає йому властивості одно сторінкового сайту. Для цього було реалізовано систему маршрутизації за допомогою React Routing.

Додаток забезпечує вибір стилю відображення секції навігації. Основний вміст сайту є динамічним, і змінюється в залежності від переходу по різним пунктам зі списку навігації. Їх було розроблено чотири:

- sales – сторінка відображення статистичних даних про продажі з певний період
- web-analytics – аналіз користування сайтом для адміністраторів. Можливість перегляду динаміки користування за різними показниками
- contacts – сторінка списку користувачів з функцією пошуку та відображення розширених даних про користувача, які можна редагувати
- orders – відображення списку замовлень, з можливістю їхньої фільтрації. Можна переглянути додаткову інформацію про замовлення

Кожен з них відображає інформацію, що є корисною для адміністраторів сайту.

|    |      |          |       |      |                           |      |
|----|------|----------|-------|------|---------------------------|------|
|    |      |          |       |      | <b>ІАЛЦ.045440.004 ПЗ</b> | Лист |
| Зм | Лист | № докум. | Підп. | Дата |                           | 51   |

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. ECMAScript [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/ECMAScript>
2. AJAX [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/AJAX>
3. JSON [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/JSON>
4. MVC [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C-%D0%B2%D0%B8%D0%B4-%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%B5%D1%80>
5. CSS [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/CSS>
6. Sass [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/Sass>
7. HTML [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/HTML>
8. HTTP [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/HTTP>
9. PHP [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/PHP>
10. Git [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/Git>
11. Frameworks [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://fructcode.com/ru/blog/features-of-popular-frameworks-html-css-php-and-python-frameworks/>



12. DOM [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://www.w3.org/DOM/>
13. W3C [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://www.w3.org/>
14. XML [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://www.w3.org/XML/>
15. ASP.NET [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/ASP.NET>
16. Who is administrator [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://kiev.hh.ua/article/22096>
17. Ubuntu [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://www.techopedia.com/definition/3307/ubuntu>
18. JavaScript [Електронний ресурс]. - Режим доступу до ресурсу:  
[https://ru.wikibooks.org/wiki/%D0%9A%D0%BD%D0%B8%D0%B3%D0%B0\\_JavaScript/%D0%9A%D1%80%D0%B0%D1%82%D0%BA%D0%B0%D1%8F\\_%D0%B8%D1%81%D1%82%D0%BE%D1%80%D0%B8%D1%8F](https://ru.wikibooks.org/wiki/%D0%9A%D0%BD%D0%B8%D0%B3%D0%B0_JavaScript/%D0%9A%D1%80%D0%B0%D1%82%D0%BA%D0%B0%D1%8F_%D0%B8%D1%81%D1%82%D0%BE%D1%80%D0%B8%D1%8F)
19. Lukas Ruebbelke with Brian Ford. AngularJS in Action. — Manning Publications, 2015.
20. React [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://www.infoworld.com/article/2608181/react--making-faster--smoother-uis-for-data-driven-web-apps.html>
21. Visual Studio Code [Електронний ресурс]. - Режим доступу до ресурсу:  
<https://code.visualstudio.com/>